

# Гиперус. Инфраструктура

Платформа управления ИТ-инфраструктурой

Руководство пользователя

---

v1.0

ООО «Гиперус» (Hyperus LLC)  
ОГРН 1233900000850, ИНН 3900006522  
[www.hyperus.ru](http://www.hyperus.ru)

© 2023-2025 ООО «Гиперус» Все авторские права защищены.

Гиперус, Hyperus и логотип Hyperus являются товарными знаками или зарегистрированными товарными знаками ООО Гиперус в России и (или) других странах.

Воспроизведение любой части данного руководства в любой форме без письменного разрешения фирмы ООО Гиперус запрещено. В настоящее руководство могут вноситься изменения без предварительного уведомления. На момент издания были внесены все изменения. Однако если будут найдены какие-либо ошибки, фирма ООО Гиперус убедительно просит сообщить ей об этом. За ошибки, обнаруженные в руководстве, фирма ООО Гиперус ответственности не несет. Не предполагается никакое обязательство за повреждения, обусловленные использованием содержащейся здесь информации.

Все наименования компаний, продуктов и услуг, логотипы, бренды и зарегистрированные или незарегистрированные товарные знаки используются только в целях идентификации, и права собственности на них принадлежат исключительно соответствующим владельцам. Использование каких-либо брендов, наименований, логотипов или любых других сведений, изображений или материалов, принадлежащих третьим лицам, не подразумевает их поддержку. Мы не являемся владельцами подобных сведений, изображений, материалов, знаков и наименований третьих лиц.

# Оглавление

<b>Аннотация</b>	<b>8</b>
Назначение ПО . . . . .	8
Традиционная ИТ-инфраструктура в закрытом контуре . . . . .	8
Инфраструктура для удалённых офисов . . . . .	8
Среда разработки и тестирования ПО . . . . .	9
Горизонтально масштабируемое ПО . . . . .	9
Интеграция с облачными сервисами . . . . .	9
Описание функционала . . . . .	10
Функциональные возможности продукта Гиперус.Инфраструктура для Пользователя . . . . .	10
<b>Введение</b>	<b>12</b>
Обзор Гиперус.Инфраструктура . . . . .	12
Аудитории документа . . . . .	12
Структура документа . . . . .	14
<b>Глава 1. Управление виртуальными машинами</b>	<b>15</b>
Шаблоны . . . . .	16
Изменение конфигурации сервера . . . . .	17
Клонирование виртуальных машин . . . . .	17
Создание снапшотов виртуальных машин . . . . .	18
Удаление виртуальных машин . . . . .	18
Управление виртуальной сетью . . . . .	18
Веб-консоль виртуальной машины . . . . .	19
Мониторинг виртуальной инфраструктуры . . . . .	19
Группа серверов . . . . .	19
Поддерживаемые политики . . . . .	19
Шаблоны конфигураций . . . . .	20
Зона доступности . . . . .	20
Образ . . . . .	21
Файловое хранилище . . . . .	21
<b>Глава 2. Графический интерфейс Гиперус.Инфраструктура</b>	<b>22</b>
Общая структура интерфейса . . . . .	22
Навигационная панель . . . . .	22

Блок - Дашборд . . . . .	23
Блок - Мониторинг и аналитика . . . . .	23
Блок - Управление виртуальными машинами . . . . .	24
Блок - Управление виртуальными сетями . . . . .	24
Блок - Управление подсистемой хранения данных . . . . .	24
Описание элементов Дашборда . . . . .	24
Элементы Мониторинг и аналитика . . . . .	25
Элементы Управление виртуальными машинами . . . . .	27
Элементы Управление виртуальными сетями . . . . .	28
Элементы Управление подсистемой хранения данных . . . . .	29
<b>Глава 3. Сеть в Виртуальных машинах</b>	<b>31</b>
Введение . . . . .	31
Пример проблемы . . . . .	31
Перед началом работы . . . . .	31
Проверьте настройки сетевого интерфейса . . . . .	32
Проверьте, что нужные приложения запущены на виртуальной машине . . . . .	35
Проверьте настройки фаервола виртуальной машины . . . . .	38
Проверьте настройки групп безопасности фаервола Гиперус . . . . .	38
Обратитесь в техническую поддержку . . . . .	40
<b>Глава 4. Порядок создания объектов vDC</b>	<b>41</b>
Начало работы . . . . .	41
Создание VM . . . . .	41
Создание конфигурации . . . . .	41
Создание образа . . . . .	41
Создание сети . . . . .	42
Непосредственное создание VM . . . . .	42
Создание зон доступности . . . . .	42
Конфигурации VM . . . . .	43
Создание конфигураций . . . . .	45
Использование CPU-pinning . . . . .	45
Использование Квот и QoS . . . . .	47
Планировщик VM . . . . .	47
Фильтрация по характеристикам поставщика ресурсов . . . . .	48
Весы . . . . .	48
Диски и СХД . . . . .	49
Бэкенды . . . . .	49
<b>Глава 5. Сеть</b>	<b>51</b>
Настройка основных компонентов . . . . .	51
Необходимые ресурсы для функционирования сетевых служб . . . . .	51

<b>Глава 6. Кластеры</b>	<b>53</b>
Введение . . . . .	53
<b>Глава 7. Подготовка VM с ОС Astra Linux</b>	<b>55</b>
Установка ОС . . . . .	55
Настройка ОС на сервере с платформой . . . . .	55
Настройка системного времени . . . . .	57
Открытый контур . . . . .	57
Закрытый контур . . . . .	58
Настройка сети на узлах кластера . . . . .	58
Минимальная конфигурация без бриджей и бондов . . . . .	58
Конфигурация с бриджем и интерфейсом во VLAN . . . . .	58
Конфигурация с бондом во VLAN в режиме active-backup . . . . .	59
Конфигурация в режиме balance-alb с VLAN . . . . .	60
Конфигурация с бондом и бриджем в режиме LACP (802.3ad) . . . . .	60
Настройка подключения к узлам кластера . . . . .	61
Уровень защищённости “Орёл” . . . . .	61
Уровень защищённости “Воронеж” . . . . .	61
<b>Решение типовых проблем пользователей ПО (FAQ)</b>	<b>63</b>
Как исправить проблему с высокой нагрузкой на кластер, приводящей к медлен- ной работе виртуальных машин? . . . . .	63
Как исправить ошибку “No valid host found” при развертывании виртуальной ма- шины? . . . . .	63
<b>Термины и сокращения</b>	<b>65</b>
Дополнительная документация на продукт Гиперус . . . . .	69
<b>Требования к составу и квалификации обслуживающего персонала</b>	<b>71</b>

# Список иллюстраций

1	Форма создание VM . . . . .	16
---	-----------------------------	----

# Список таблиц

1	Шаблоны конфигурации VM . . . . .	20
---	-----------------------------------	----

# Аннотация

Данный документ содержит описание базовых операций пользователей ПО Гиперус.Инфраструктура и призван помочь пользователям эффективнее использовать ПО. Документ предназначен для инженеров группы эксплуатации и сопровождения продукта, которые используют Гиперус.Инфраструктура.

## Назначение ПО

Традиционная ИТ-инфраструктура в закрытом контуре, инфраструктура для удалённых офисов, среда разработки и тестирования ПО, горизонтально масштабируемое ПО.

ПО Гиперус.Инфраструктура предназначено для создания, управления и эффективного использования гиперконвергентной инфраструктуры в различных сценариях применения. Обеспечивает унифицированное решение для задач, связанных с вычислениями, хранением и сетевым взаимодействием, позволяя организациям адаптироваться к быстрым изменениям в бизнес-среде. Ниже приведены ключевые аспекты назначения комплекса в различных контекстах.

### Традиционная ИТ-инфраструктура в закрытом контуре

В традиционной инфраструктуре в закрытом контуре организации используют физические серверы и отдельные системы хранения данных. Гиперус.Инфраструктура упрощает управление такими системами, интегрируя вычислительные ресурсы и хранилище в единую платформу. Гиперус.Инфраструктура дает возможность:

- Сократить общие затраты на оборудование и управление им;
- Упростить администрирование с помощью централизованной панели управления;
- Повысить надежность за счет резервирования и автоматического распределения ресурсов.

### Инфраструктура для удалённых офисов

С увеличением числа удалённых офисов организации сталкиваются с необходимостью создать безопасные и удобные рабочие пространства для сотрудников. Гиперус.Инфраструктура дает возможность:



- Реализовывать защищённый удалённый доступ к корпоративным ресурсам.
- Масштабировать инфраструктуру в зависимости от роста бизнеса и количества удалённых пользователей.
- Обеспечивать высокую доступность и производительность приложений независимо от местоположения.

## **Среда разработки и тестирования ПО**

Создание сред разработки и тестирования требует высоких темпов развертывания и управления. Гиперус.Инфраструктура дает возможность:

- Быстрого создания и уничтожения виртуальных машин для тестирования различных сценариев.
- Клонирования и создания снэпшотов для легкого возврата к предыдущим версиям и быстрого тестирования изменений.
- Интеграции с CI/CD (непрерывной интеграцией и поставкой), позволяя улучшить совместную работу команд и автоматизацию процессов.

## **Горизонтально масштабируемое ПО**

Современные приложения требуют высокой производительности и горизонтальной масштабируемости для обработки больших объемов данных. Гиперус.Инфраструктура дает возможность:

- Гибко увеличивать вычислительные ресурсы путём добавления новых узлов в кластер.
- Контейнеризации и оркестрации приложений для оптимизации развертывания и управления.
- Упрощения работы с микросервисами, значительно ускоряя внедрение новых функций и инноваций.

## **Интеграция с облачными сервисами**

Гиперус.Инфраструктура легко интегрируется с облачными решениями, предлагая организациям возможность:

- Создавать гибридные архитектуры, которые объединяют локальные и облачные ресурсы.
- Использовать облачные услуги по мере необходимости, оптимизируя затраты на хранение и вычисления.
- Быстро масштабироваться для обработки всплесков нагрузки.

## Описание функционала

Гиперконвергентная платформа Гиперус.Инфраструктура предназначена для создания и управления программно-определяемых центров обработки данных внутри периметра организации. Основа для создания программно-аппаратного комплекса Гиперус.

Гиперус.Инфраструктура реализует концепцию программно-определяемого центра обработки данных, который предоставляет простой и унифицированный доступ к различным вычислительным ресурсам, сетям передачи данных, системам хранения данных, а также дополнительным сервисам, таким как:

- балансировщики нагрузки (Load Balancer as a Service),
- средства защиты периметра (Firewall as a Service, Security Groups),
- объектное хранение данных, совместимое с Amazon S3.

Портал Управления Инфраструктурой предоставляет централизованный доступ к ресурсам платформы Гиперус.Инфраструктура. Пользователи с ролью Администратор виртуального датацентра имеет возможность создавать виртуальные сети, маршрутизаторы, дисковые устройства, а платформа Гиперус.Инфраструктура внедряет эти изменения на низком уровне, обеспечивая интеграцию с сетевым оборудованием и системами хранения данных.

После настройки сетей пользователям и администраторам доступна возможность создавать виртуальные серверы из заранее заданных образов виртуальных серверов. Пользовательские операции не требуют знания используемых низкоуровневых технологий.

Гиперус.Инфраструктура взаимодействует с базовой инфраструктурой через открытые или предоставляемые вендором драйверы, что помогает спасти клиентов от ловушки конкретной технологии, вендора или инструмента.

Гиперус.Инфраструктура предоставляет дополнительные услуги, такие как управление идентификацией, оркестрация, учет потребляемых ресурсов в той же программной основе через API. Гиперус.Инфраструктура также предлагает основу для эволюции к DevOps, непрерывной интеграции и методологии непрерывного развертывания.

## Функциональные возможности продукта Гиперус.Инфраструктура для Пользователя

- Создание и управление виртуальными машинами: Пользователь может создавать новые виртуальные машины, устанавливать необходимые операционные системы и конфигурировать их под свои нужды.
- Мониторинг состояния ресурсов: Пользователь имеет доступ к информации о состоянии своих виртуальных машин, включая загрузку процессора, использование памяти и дискового пространства, что позволяет оперативно реагировать на изменения.

- **Настройка сети:** Возможность подключать виртуальные машины к различным сетям, настраивать IP-адресацию и управлять правилами безопасности для обеспечения защиты данных.
- **Управление хранилищем:** Пользователь может добавлять и управлять виртуальными дисками, изменять их размеры и конфигурации, а также выполнять резервное копирование данных.
- **Оптимизация производительности:** Возможность управлять ресурсами виртуальных машин, например, выделять больше оперативной памяти или процессорных ядер в ответ на повышенные требования приложений.
- **Интерактивный доступ к виртуальным машинам:** Пользователь может подключаться к своим виртуальным машинам через графический интерфейс или удаленный доступ для выполнения необходимых операций.
- **Создание шаблонов виртуальных машин:** Пользователь может создавать шаблоны для быстрого развертывания новых виртуальных машин с заранее установленными настройками и конфигурациями.
- **Анализ производительности:** Возможность получения отчетов о производительности виртуальных машин и ресурсах, что позволяет пользователю лучше понимать использование своих ресурсов.
- **Поддержка и документация:** Доступ к полному набору справочных материалов и документации, которые помогают пользователю разобраться с функциональными возможностями платформы и решать возникающие вопросы.
- **Обратная связь с администратором:** Возможность отправлять запросы на изменения или разрешение проблем с системными ресурсами, обеспечивая эффективное взаимодействие с администратором системы.

# Введение

## Обзор Гиперус.Инфраструктура

Гиперус.Инфраструктура – системное программное обеспечение для создания программно-определяемой виртуальной инфраструктуры в составе программно-аппаратного комплекса Гиперус. Гиперус.Инфраструктура построен на принципе гиперконвергенции. Гиперконвергенция это принцип, в котором аппаратные составляющие физических серверов реализуются через программно-определяемые слои (Хранилище, Сеть, Виртуализация).

Ключевые особенности Гиперус.Инфраструктура включают:

- Быстрый запуск и развертывание инфраструктуры;
- Готовые инструменты для управления виртуальным дата-центром и сетевыми функциями;
- Блочное хранение с настраиваемой производительностью;
- Авторизация и аутентификация через отдельный модуль;
- Интеллектуальная оптимизация ресурсов для повышения производительности;
- Возможность интеграции с существующими системами;
- Энергоэффективный подход, поддерживающий экосознательность;
- Техническая поддержка и возможность индивидуально адаптировать решения для конкретного партнера.

## Аудитории документа

**Системные администраторы** - Отвечающие за установку, настройку и поддержку Гиперус.Инфраструктура в организациях. Технические знания и опыт работы с виртуализацией и сетевой инфраструктурой.

*Цели:* Эффективное управление ресурсами, решение проблем и оптимизация производительности.

*Компетенции:*

- CompTIA Linux+;
- Red Hat Certified System Administrator (RHCSA);
- Microsoft Certified: Azure Administrator Associate;
- Certified Kubernetes Administrator (CKA);

- Опыт работы с сетевыми протоколами и инструментами мониторинга.

**Инженеры по облачным технологиям** - Специалисты, занимающиеся проектированием и внедрением облачных решений на основе Гиперус.Инфраструктура. Работают над интеграцией облачных и локальных ресурсов.

*Цели:* Гибридные развертывания, управление облачными сервисами и оптимизация расходов.

*Компетенции:*

- AWS Certified Solutions Architect;
- Google Cloud Professional Cloud Architect;
- Microsoft Certified: Azure Solutions Architect Expert;
- OpenStack Certified Administrator;
- Опыт в разработке и интеграции API.

**IT-менеджеры и руководители** - Специалисты, принимающие решения о развитии IT-инфраструктуры в организации. Для понимания возможностей Гиперус.Инфраструктура, стратегического планирования и бюджета.

*Цели:* Оценка ROI после внедрения Гиперус.Инфраструктура, управление ресурсами и планирование IT-стратегий.

**Разработчики** - Пользователи, работающие с приложениями, которые развертываются на платформе Гиперус.Инфраструктура. Заинтересованы в автоматизации развертывания и интеграции с другими приложениями и сервисами.

*Цели:* Использование API Гиперус.Инфраструктура, автоматизация процессов и интеграция с облачными приложениями.

**Служба технической поддержки** - Специалисты, работающие в команде поддержки пользователей Гиперус.Инфраструктура. Они должны иметь доступ к информации о функциональности и настройках для эффективного решения проблем.

*Цели:* Повышение качества обслуживания, быстрое решение проблем и предоставление точной информации пользователям.

**Пользователи конечных приложений** - Сотрудники организаций, использующие приложения, развернутые на платформе Гиперус.Инфраструктура. Хотя их технические знания могут быть ограничены, они должны понимать, как работать с интерфейсом и решать базовые проблемы.

*Цели:* Эффективное использование приложений и обратная связь с IT-отделом для улучшения работы.

## Структура документа

Руководство пользователя описывает графический web-интерфейс, объекты и возможности платформы.

В описании документа используются информационные вставки трех видов:

### Обратите внимание

Критически важная информация для работы с системой.

### Рекомендация

Рекомендация по работе с системой в рамках какой-либо инструкции.

### Информация

Дополнительная информация к описываемому блоку информации.

# Глава 1. Управление виртуальными машинами

Работа с виртуальным сервером начинается с установки операционной системы – Windows или Unix. Но установка операционной системы отличается от установки обычного, «железного» сервера.

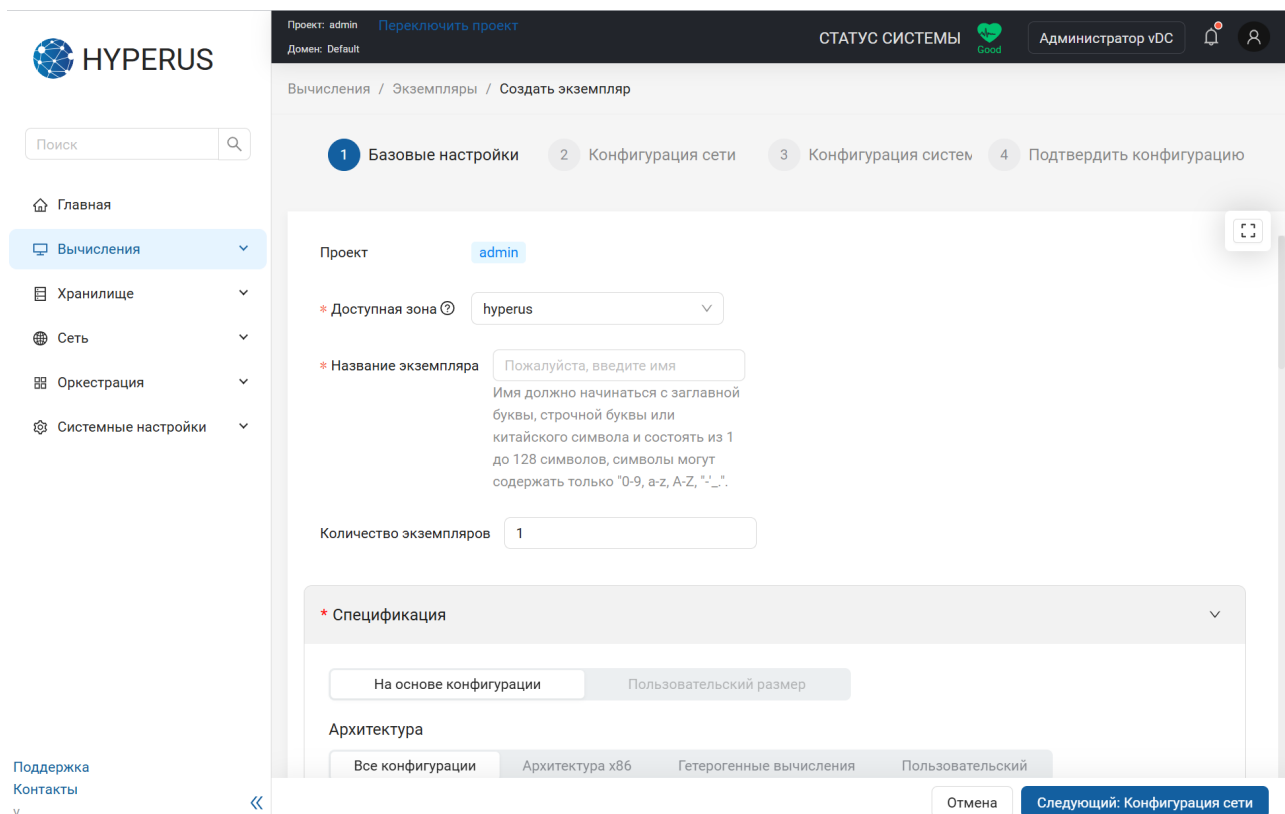
Операционную систему невозможно установить с обычного CD или DVD-диска или загрузочной флешки. Операционную систему можно установить из ISO-файла, то есть специального файла, содержащего образ загрузочного диска. Создать такой файл из обычного загрузочного DVD-диска можно при помощи специальных программ. Затем этот файл загружается в хранилище администратором. Во время установки операционной системы на виртуальную машину можно выбрать этот файл из хранилища.

Виртуальная машина (VM) — совокупность ресурсов, которые эмулируют поведение реального компьютера. Для своей работы VM использует ресурсы физического сервера (хоста). На одном хосте может размещаться несколько VM.

Виртуальные машины создаются при помощи мастера, который предлагает последовательность шагов. Во время работы мастера заказчик задает название машины и указывает необходимые характеристики:

[[@todo я бы ввел vCPU, vRAM и другие слова с приставкой “v”, чтобы пользователю было проще в тексте отличить виртуальные сущности от физических. я о том, что в абзаце ниже есть vCPU и vRAM, но за ними идут “тип и размер” диска, то есть уже отсутствует приставка “v”, а значит это физические диски? нет? тогда надо явно об этом писать)

- количество виртуальных процессоров (vCPU);
- объем оперативной памяти (vRAM);
- тип и размер диска;
- зона доступности;
- количество сетевых интерфейсов;
- операционная система.



**Рис. 1:** Форма создание VM

## Шаблоны

Другой способ установки операционной системы – шаблоны. Шаблон – это готовая к работе операционная система и приложения, необходимые для решения задач заказчика.

Например, чтобы создать веб-сервер на виртуальной машине, необходимо выполнить несколько шагов:

- установить операционную систему;
- установить необходимые службы, например систему управления базами данных;
- включить поддержку таких языков программирования, например PHP или Python;
- установить службу веб-сервера;
- закачать содержимое веб-сайта.

Шаблон же содержит все необходимые компоненты веб-сервера, и после разворачивания виртуальной машины из шаблона заказчику остается только добавить в веб-сервер содержимое веб-сайта.

Шаблоны могут быть созданы Администратором инсталляции или самим заказчиком. Шаблоны делятся на категории по сферам использования:

- Сервера приложений:
  - Предустановленные серверы приложений, таких как веб-серверы.



- Администрирование:
  - Шаблоны, для поддержки сетевого взаимодействия, управление рабочими станциями и серверами,
  - Системы мониторинга сети, серверов и приложений.
- Безопасность и сетевое взаимодействие:
  - Предустановленные средства по защите сети (брандмауэры), антиспам фильтры или антивирусы.
- Серверы баз данных и почтовые серверы:
  - Эти шаблоны включают в себя бесплатные серверы баз данных и почтовые серверы, уже снабженные средствами безопасности. Серверы можно использовать с другими системами, расположенными в виртуальной инфраструктуре, например, с веб-приложением или антиспам фильтром, также созданным из шаблона.
- Операционные системы:
  - Шаблоны содержат настроенные виртуальные машины с установленной операционной системой Windows Server или Linux.

## Изменение конфигурации сервера

Пользователь через Панель Управления имеет возможность:

- добавлять или уменьшить размер оперативной памяти, число ЦПУ, увеличить размер дисков;
- выбирать тип накопителя при создании виртуальной машины;
- переносить данные с одного накопителя на другой, без необходимости перезагружать сервер;
- подключать дополнительные диски к VM;
- изменять тип или версию ОС при создании VM.

## Клонирование виртуальных машин

В панели управления имеется возможность клонировать серверы, то есть создавать копии имеющихся виртуальных машин, без необходимости устанавливать операционную системы из шаблона или дистрибутива. При помощи клонирования можно быстро получить необходимое количество виртуальных серверов, например, для создания отказоустойчивого кластера.

## Создание снимков виртуальных машин

Снимок (snapshot) – это мгновенный снимок виртуальной машины. Такие снимки позволяют быстро вернуть VM в то состояние, в котором он был в момент создания снимка. Это позволяет тестировать изменения в виртуальной машине, и быстро вернуть все как было в случае, если что-то пойдет не так.

## Удаление виртуальных машин

После того как жизненный цикл проекта подошел к концу, или принято решение об удалении виртуальной машины из виртуальной инфраструктуры, ненужную виртуальную машину нужно удалить. Удаление виртуальной машины – неотменяемая и необратимая операция, восстановить удаленную машину невозможно. Если к виртуальной машине были подключены созданные ранее диски, то при удалении виртуальной машины эти диски будут отключены и сохранены. Данные на этих дисках останутся, и в дальнейшем эти диски можно будет подключить к другой виртуальной машине.

## Управление виртуальной сетью

После создания виртуальных машин в панели управления можно соединить между собой при помощи внутренней сети с необходимой пропускной способностью – от 1 Гб/с до 10 Гб/с, или дать возможность принимать запросы через Интернет.

Гиперус.Инфраструктура позволяют строить инфраструктуры любой сложности – от соединения двух серверов до многоуровневой корпоративной инфраструктуры, состоящей из частных и публичных сетей и подсетей.

Например, облачную сеть можно разделить следующим образом:

- Общая публичная сеть;
- Серверы в общей публичной сети доступны из сети Интернет;
- Клиентская публичная сеть;
- Серверы в публичной сети доступны из сети Интернет, и им выдается блок IP-адресов;
- Клиентская частная сеть;
- Серверы в клиентской частной сети не доступны из сети Интернет, они доступны только для персонала.

В панели управления можно настроить различные параметры виртуального брандмауэра, который будет ограничивать запросы к вашей виртуальной инфраструктуре.

## Веб-консоль виртуальной машины

Веб-консоль позволяет видеть экран виртуальной машины до загрузки и во время загрузки операционной системы. Это позволяет управлять сервером даже при отсутствии сетевого соединения с виртуальной машиной и восстанавливать работоспособность сервера в случае аварии. При помощи веб-консоли можно также войти в BIOS виртуальной машины и настроить там порядок загрузки операционной системы.

## Мониторинг виртуальной инфраструктуры

Мониторинг виртуальной инфраструктуры позволяет собирать и хранить различные показатели по производительности виртуальных машин. Эти показатели включают в себя данные по использованию процессора, памяти, дисков, показатели задержек при операциях с дисками и прочую информацию.

## Группа серверов

Группа серверов — совокупность серверов внутри кластера, которые объединяются в соответствии с определенной политикой. Под сервером подразумеваются ресурсы:

- виртуальные машины;
- инстансы БД;
- кластеры K8s;
- инстансы приложений из Магазина приложений.

## Поддерживаемые политики

Политика определяет размещение ресурсов на вычислительных узлах платформы (compute nodes).

- **affinity** — размещать VM на одном вычислительном узле;
- **anti-affinity** — размещать VM на разных вычислительных узлах;
- **soft-affinity** — размещать VM по возможности на одном вычислительном узле (если места на одном узле недостаточно, вместо ошибки создания, как в случае с **affinity**, будет создан сервер на другом вычислительном узле);
- **soft-anti-affinity** — размещать VM по возможности на разных вычислительных узлах. Пример использования группы серверов: развернуть несколько VM приложений только на одном узле, чтобы ускорить связь между этими VM.

## Шаблоны конфигураций

На платформе Гиперус.Инфраструктура количество процессоров и объем оперативной памяти VM задаются с помощью шаблонов конфигураций.

**Таблица 1:** Шаблоны конфигурации VM

Название	Параметры конфигурации	Описание
Basic	До 2 vCPU До 4 ГБ RAM	Базовые конфигурации для создания VM с невысокой производительностью
Standard	От 2 до 4 vCPU От 4 ГБ до 16 ГБ RAM	Конфигурации с увеличенным количеством CPU и объемом RAM
Advanced	От 8 до 16 vCPU От 16 ГБ до 64 ГБ RAM	Конфигурации для создания высокопроизводительных VM
Heavy	От 16 vCPU От 64 ГБ RAM	Индивидуальные конфигурации для создания высокопроизводительных VM

## Зона доступности

Зона доступности — это один или несколько центров обработки данных (ЦОД), в которых могут быть размещены компоненты облачной инфраструктуры. Каждая зона изолирована от сбоев в других зонах доступности. Размещение виртуальных ресурсов в нескольких зонах обеспечивает отказоустойчивость и снижает вероятность потери данных.

Гиперус.Инфраструктура распределяет по поддерживаемым зонам доступности следующие виртуальные ресурсы:

- виртуальные машины;
- диски: размещаются в сетевом блочном хранилище и автоматически реплицируются внутри своей зоны доступности;
- балансировщики нагрузки.

## Образ

Конфигурация и данные виртуальной машины хранятся в виде одного или нескольких файлов. Эти файлы могут использоваться как образ для запуска VM в локальной среде или для создания новой VM. На платформе Гиперус.Инфраструктура можно создать образ на основе диска VM или загрузить файл образа. Полученный образ можно использовать для создания новой виртуальной машины. Образ, источником которого является диск, позволяет создать копию виртуальной машины с установленной и настроенной операционной системой и приложениями. Образ, загруженный из файла, можно использовать для миграции VM или для VM машины с вашей операционной системой.

## Файловое хранилище

Файловое хранилище — это виртуальная файловая система, которую можно подключить к нескольким виртуальным машинам и использовать для совместного доступа к данным. При создании файлового хранилища указывается протокол сетевого доступа к нему (NFS или CIFS).

# Глава 2. Графический интерфейс Гиперус.Инфраструктура

В этой главе представлено подробное описание графического интерфейса Гиперус.Инфраструктура, который обеспечивает пользователям удобный и интуитивно понятный доступ к функциональности платформы. Интерфейс разработан с акцентом на простоту использования и эффективность, что позволяет администраторам и пользователям быстро выполнять необходимые действия.

## Общая структура интерфейса

Графический интерфейс Гиперус.Инфраструктура состоит из нескольких основных компонентов:

- **Навигационная панель:** Расположена слева, эта панель позволяет пользователям легко перемещаться между различными секциями системы, такими как “Кластеры”, “Виртуальные машины”, “Хранилище” и “Мониторинг”. Каждый раздел представлен в виде иконки и названия, что упрощает поиск нужной функции.
- **Основная область содержимого:** Здесь отображается информация, связанная с выбранной секцией. Пользователи могут просматривать данные, выполнять операции над виртуальными машинами и кластерами, а также получать сведения о состоянии ресурсов.
- **Вспомогательные элементы управления:** Эти элементы включают кнопки для выполнения основных действий, таких как “Создать”, “Изменить” и “Удалить”, а также фильтры и параметры поиска, которые помогают быстро находить нужные элементы.

## Навигационная панель

Данная панель расположена в левой части окна и представляет собой дерево всех объектов и связей, расположенных в иерархическом порядке (Рисунок). Панель навигации отображает актуальную информацию о текущих состояниях всех объектов, выделяя их соответствующим цветом. Соответствие цвета и состояния можно посмотреть в легенде в правом нижнем углу экрана. При выборе объекта происходит переход на него - на главном экране для выбранного объекта открывается топология вложенных объектов. Если вложенных объектов нет, то открывается конечный вид объекта - подробная информация по объекту.

## Блок - Дашборд

Дашборд является стартовой страницей web-интерфейса Гиперус.Инфраструктура и предоставляет сводную информацию о состоянии виртуальной инфраструктуры.

На дашборде можно увидеть основные показатели производительности, включая:

- **Использование ресурсов:** Отображает текущую загрузку CPU, памяти и дискового пространства. Пользователь может оценить, насколько эффективно используются ресурсы и где могут возникнуть узкие места.
- **Состояние кластеров:** Предоставляет информацию о состоянии всех кластеров в системе, включая их доступность, количество активных и неактивных узлов. Это помогает следить за общей стабильностью инфраструктуры.
- **Активные виртуальные машины:** Список всех виртуальных машин с указанием их текущего состояния (включена/выключена), а также распределение по различным кластерам и хостам. Это позволяет пользователю быстро определить, где находятся потенциальные проблемы.
- **Графики и диаграммы состояния:** Интерактивные визуализации, позволяющие быстро оценить текущее состояние системы. Например, могут быть представлены:
  - Графики нагрузки по временным интервалам;
  - Диаграммы использования памяти и постоянного хранилища;
  - Визуализация сетевой активности.
- **Уведомления и оповещения:** Дашборд также может показывать текущее состояние предупреждений и уведомлений о проблемах или неисправностях в системе, что позволяет пользователю быстро реагировать на потенциальные риски.
- **Скорость отклика приложений:** Графики, показывающие время отклика приложений, развернутых на виртуальных машинах, что помогает пользователям оценивать производительность своей инфраструктуры в реальном времени.
- **Фильтрация и настройки отображения:** Пользователь может настраивать видимость определенных показателей или фильтровать графики по типу ресурсов, времени и другим критериям для более точного анализа.

## Блок - Мониторинг и аналитика

Раздел мониторинга предоставляет детальную информацию о производительности и состоянии ресурсов. Пользователи могут настраивать отображаемую информацию, выбирая различные метрики (например, использование процессора, памяти и дискового пространства) и временные интервалы. Функция уведомлений позволяет получать оповещения о критических состояниях системы в реальном времени.

## **Блок - Управление виртуальными машинами**

Интерфейс предлагает простой и понятный процесс управления виртуальными машинами. Пользователи могут создавать, настраивать и удалять виртуальные машины, а также изменять их конфигурации. Раздел управления виртуальными машинами включает в себя вкладки для настройки сети, хранилища и параметров ресурсов, что делает процесс управления многофункциональным и гибким.

## **Блок - Управление виртуальными сетями**

Раздел “Управление виртуальными сетями” в интерфейсе Гиперус.Инфраструктура предназначен для настройки, мониторинга и управления сетевыми ресурсами в виртуальной инфраструктуре. Он предоставляет пользователям инструменты для создания безопасных и эффективных сетевых решений, необходимых для работы виртуальных машин и приложений.

## **Блок - Управление подсистемой хранения данных**

В данном разделе описаны основные функции управления подсистемой хранения данных в Гиперус, предназначенные для конечных пользователей. Интерфейс предоставляет необходимые инструменты для управления хранилищем, что позволяет повышать производительность и обеспечивать безопасность данных.

## **Описание элементов Дашборда**

Дашборд в интерфейсе Гиперус.Инфраструктура представляет собой центральную компоненту, обеспечивающую пользователям доступ к быстро усваиваемой и актуальной информации о состоянии виртуальной инфраструктуры. Он служит отправной точкой для администрирования, предоставляя все необходимые данные в одном месте.

### **Сводная информация:**

- **Показатели производительности:** Дашборд отображает ключевые метрики, такие как использование CPU, памяти и дискового пространства для каждого кластера и виртуальной машины. Эти данные позволяют быстро оценить загрузку ресурсов и выявить возможные узкие места.
- **Статус кластеров:** Визуальные индикаторы (например, цветовые коды или значки) показывают состояние кластеров (нормальное, предупреждение, критическое). Это помогает мгновенно определить, где требуется внимание.

### **Графики и диаграммы:**



- Тренды производительности: Линейные и столбчатые графики иллюстрируют изменения в использовании ресурсов во времени. Это позволяет администраторам видеть, как нагрузка на систему меняется, и планировать масштабирование инфраструктуры.
- Сравнительный анализ: Возможность отображения данных по разным кластерам или виртуальным машинам для сравнения производительности и идентификации наилучших практик.

### **Функции быстро доступа:**

- Кнопки для выполнения операций: На дашборде присутствуют кнопки для быстрого создания новых виртуальных машин, управления кластерами и запуска основных задач. Это значительно упрощает работу администраторов, позволяя выполнять действия без необходимости перехода в другие разделы интерфейса.
- Фильтры и настройки отображения: Пользователи могут настраивать, какие данные отображать на дашборде, используя фильтры по времени, типам ресурсов или статусам. Это позволяет индивидуализировать интерфейс в соответствии с потребностями конкретной задачи.

### **Уведомления и оповещения:**

- **Информационные сообщения:** Дашборд отображает важные уведомления о состоянии системы, например, о выполнении запланированных задач, ошибках или предупреждениях. Эти сообщения позволяют администраторам быстро реагировать на критические ситуации.

### **Интерактивность:**

- **Динамическое обновление данных:** Дашборд обеспечивает автоматическое обновление показателей, что позволяет пользователям всегда видеть актуальную информацию без необходимости вручную обновлять страницу.
- **Детализация информации:** При нажатии на графики или метрики пользователи могут развернуть детальную информацию, чтобы глубже изучить данные и выявить причины проблем.

## **Элементы Мониторинг и аналитика**

Раздел “Мониторинг и аналитика” в интерфейсе Гиперус.Инфраструктура предоставляет пользователям инструменты для отслеживания состояния виртуальной инфраструктуры и анализа производительности ресурсов. Этот раздел разработан для того, чтобы администраторам было легко выявлять потенциальные проблемы, оптимизировать производительность и принимать обоснованные решения на основе фактических данных.

### **Общий обзор состояния системы:**

- **Интерактивные панели:** Пользователи могут видеть общую картину состояния кластера и виртуальных машин через компактные виджеты с ключевыми метриками, такими как загрузка CPU, использование памяти и дискового пространства. Эти панели помогают быстро оценить текущее состояние системы.

#### **Детализированные показатели:**

- **Метрики производительности:** Доступ к важной информации, такой как время отклика, пропускная способность сети и уровень ввода/вывода дисков. Эти метрики позволяют отслеживать производительность и выявлять узкие места в инфраструктуре.
- **Исторические данные:** Возможность анализа производительности за разные временные интервалы (часы, дни, недели) позволяет пользователям выявлять тенденции и аномалии в работе системы.

#### **Графики и визуализация данных:**

- **Динамические графики:** Пользователи могут взаимодействовать с графиками, чтобы получать более глубокий анализ показателей, таких как использование ресурсов по времени или в зависимости от рабочих нагрузок. Удобные средства визуализации помогают лучше понимать сложные данные.
- **Сравнительные диаграммы:** Функции, позволяющие сравнивать состояние разных узлов и виртуальных машин, что позволяет быстро определить, какие элементы инфраструктуры работают эффективно, а какие требуют внимания.

#### **Уведомления и оповещения:**

- **Настраиваемые триггеры:** Пользователи могут настраивать систему уведомлений по критическим метрикам, например, превышению порогов использования ресурсов. Это гарантирует, что администраторы своевременно получают важные уведомления, позволяя предотвратить потенциальные проблемы.
- **Исторические события и журналы:** Запись всех значимых событий в системе, включая изменения конфигураций и операции, помогает пользователям анализировать прошлые события и принимать решения на основе исторических данных.

#### **Аналитические инструменты:**

- **Отчеты о производительности:** Система создает автоматизированные отчеты, которые помогают в оценке производительности инфраструктуры и выявлении долгосрочных трендов. Пользователи могут настраивать отчеты под свои нужды и экспортировать их для дальнейшего анализа.
- **Обработка данных в реальном времени:** Мониторинг инфраструктуры анализируют данные в реальном времени, обеспечивая пользователям актуальную информацию для мгновенного реагирования на изменения.

## Элементы Управление виртуальными машинами

Раздел “Управление виртуальными машинами” в интерфейсе Гиперус.Инфраструктура предоставляет пользователям набор инструментов для создания, настройки и мониторинга виртуальных машин (VM). Этот раздел разработан с учетом потребностей администраторов и пользователей, обеспечивая гибкость и удобство в управлении ресурсами виртуальной инфраструктуры.

### Создание и развертывание виртуальных машин:

- **Мастер установки:** Интуитивно понятный процесс создания виртуальной машины с помощью мастера, который пошагово ведет пользователя. Включает выбор операционной системы, выделение ресурсов (CPU, память, дисковое пространство) и настройку сетевых параметров.
- **Шаблоны:** Возможность создания шаблонов для быстрого развертывания стандартных конфигураций виртуальных машин, что значительно ускоряет процессы настройки и развертывания.

### Настройка параметров виртуальных машин:

- **Компоненты и ресурсы:** Установка и изменение параметров виртуальной машины, таких как количество виртуальных процессоров, объем оперативной памяти, типы дисков и конфигурация сети.
- **Политики и теги:** Использование политик для управления производительностью, резервным копированием и безопасностью. Применение тегов для упрощения поиска и фильтрации виртуальных машин по различным критериям.

### Мониторинг и управление состоянием VM:

- **Статус и производительность:** Визуальное отображение текущего состояния каждой виртуальной машины, включая использование CPU, памяти, дисков и сети. Администраторы могут быстро обнаруживать потенциальные проблемы и принимать меры для их устранения.
- **Управление жизненным циклом:** Поддержка операций, таких как запуск, остановка, приостановка, перезагрузка и удаление виртуальных машин, обеспечивая гибкость в управлении ресурсами.

### Миграция виртуальных машин:

- **Горячая миграция:** Возможность перемещать виртуальные машины между узлами кластера без остановки их работы. Это значительно повышает доступность и позволяет эффективно распределять нагрузку.

### Резервное копирование и восстановление:

- **Инструменты для резервного копирования:** Легкие в использовании функции для создания резервных копий виртуальных машин, а также опции для восстановления на

базе точек восстановления. Это обеспечивает безопасность данных и быстрый откат при необходимости.

- **Политики резервного копирования:** Настройка регулярных расписаний резервного копирования, позволяющая автоматизировать этот процесс и гарантировать постоянное сохранение данных.

## Элементы Управление виртуальными сетями

Основные возможности управления виртуальными сетями.

- **Настройка виртуальных локальных сетей (VLAN):** Пользователи могут создавать и управлять виртуальными локальными сетями для разделения трафика и повышения безопасности. Поддержка VLAN позволяет гибко управлять сетевыми конфигурациями и сегментировать трафик между различными виртуальными машинами.
- **Определение сетевых политик:** Возможность применения сетевых политик для управления правилами трафика и доступа, обеспечивая оптимальную безопасность виртуальной инфраструктуры.

### Управление IP-адресацией:

- **Динамическое и статическое распределение IP-адресов:** Поддержка как динамического (DHCP), так и статического назначения IP-адресов для виртуальных машин, что обеспечивает гибкость в управлении адресным пространством.
- **Автоматизированная конфигурация:** Инструменты для автоматической настройки IP-адресов и других сетевых параметров, что упрощает процесс развертывания и управления сетью.

### Мониторинг сетевого трафика:

- **Анализ производительности сети:** Интерактивные инструменты мониторинга показывают использование сетевых ресурсов, такие как пропускная способность, задержка и уровень потерянных пакетов. Это помогает администраторам быстро выявлять узкие места и аномалии в работе сети.
- **Сетевые оповещения:** Уведомления о критических событиях в сети, таких как превышение установленных лимитов по пропускной способности или ошибки в передаче данных, что позволяет оперативно реагировать на проблемы.

### Безопасность сетей:

- **Настройка правил доступа:** Функции для определения правил брандмауэра, фильтрации трафика и управления доступом между виртуальными машинами, что защищает инфраструктуру от несанкционированного доступа.
- **VPN и шифрование:** Инструменты для настройки виртуальных частных сетей (VPN) и шифрования трафика, обеспечивающие безопасное подключение между удаленными пользователями и ресурсами виртуальной инфраструктуры.

### **Интеграция с облачными сетями:**

- **Гибридная сеть:** Возможность подключения виртуальных сетей к облачным ресурсам для создания гибридных решений, что позволяет использовать мощности как локальной, так и облачной инфраструктуры.
- **Синхронизация с облачными провайдерами:** Поддержка интеграции с различными облачными платформами для управления сетевыми компонентами в облаке, что упрощает развертывание и управление расширенным окружением.

### **Автоматизация сетевых операций:**

- **API и скрипты:** Возможность автоматизации управления сетью с использованием API, позволяющая писать собственные скрипты для выполнения рутинных задач и интеграции с другими системами управления.

## **Элементы Управление подсистемой хранения данных**

Основные функции управления хранилищем.

### **Создание и управление хранилищем:**

- **Выбор типа хранилища:** Вы можете создать хранилище, выбрав подходящий тип – локальное, сетевое (например, iSCSI или NFS) или облачное. Это позволяет оптимально настроить хранение данных в зависимости от ваших потребностей.
- **Настройка LUN (логических юнитов):** Вы можете создать логические единицы хранения (LUN), которые распределяют объем хранилища для ваших приложений или виртуальных машин. Это позволяет более эффективно организовать использование доступных ресурсов.

### **Мониторинг состояния и производительности:**

- **Дашборды и отчеты:** В интерфейсе есть интуитивно понятные дашборды с ключевыми метриками, такими как использование дискового пространства, скорость чтения/записи и задержки. Это позволяет вам быстро оценить текущее состояние хранилища и выявить возможные проблемы.
- **Уведомления и оповещения:** Вы будете получать уведомления о критических событиях, таких как недостаток свободного места или проблемы с доступом к данным, что позволит вовремя реагировать на возможные неполадки.

### **Оптимизация использования хранилища:**

- **Дедупликация и сжатие данных:** Гиперус.Инфраструктура поддерживает сжатие данных, что помогает снизить объем используемого дискового пространства и уменьшает затраты на хранение. Вы можете легко активировать или деактивировать эти функции в зависимости от ваших нужд.

- **Клонирование и снимки:** Вы можете создавать мгновенные снимки (снимки) текущего состояния виртуальных машин и хранилищ, что обеспечивает возможность быстрого восстановления данных в случае необходимости. Клонирование виртуальных машин упрощает процесс разработки и тестирования приложений.

#### **Резервное копирование и восстановление:**

- **Настройка автоматического резервного копирования:** Вы можете настроить регулярные автоматические резервные копии данных, что обеспечивает защиту информации от потерь. Выбор расписания и уровня резервных копий доступен прямо в интерфейсе.
- **Гибкие опции восстановления:** В случае повреждения данных или других сбоев Гиперус.Инфраструктура предлагает простые в использовании инструменты восстановления, чтобы вернуть данные в работоспособное состояние.

#### **Безопасность данных:**

- **Аутентификация и контроль доступа:** Вы можете настроить учетные записи пользователей и политики доступа, что поможет защитить данные от несанкционированного доступа. Поддержка шифрования данных обеспечивает дополнительную защиту для конфиденциальной информации.
- **Журналирование и аудит:** Система ведет журналы всех операций с данными, что позволяет вам отслеживать действия и обеспечивать прозрачность операций в соответствии с политиками безопасности.

# Глава 3. Сеть в Виртуальных машинах

## Введение

В данном разделе приведены рекомендации по устранению проблем с конфигурацией сети на виртуальных машинах Linux.

В качестве примера предполагается, что:

- виртуальная машина имеет только один сетевой интерфейс;
- решается проблема отсутствующего доступа к виртуальной машине по SSH.

Описанные подходы также могут быть использованы, когда доступ потерян до другого приложения (например, веб-сервера NGINX) или когда у виртуальной машины есть несколько сетевых интерфейсов.

## Пример проблемы

- Невозможно подключиться к виртуальной машине по SSH по ее приватному IP-адресу или плавающему IP-адресу (если он есть).
- Некоторое время назад проблем с подключением не было.

## Перед началом работы

1. Выполните принудительную перезагрузку.

Эта операция может помочь, если виртуальная машина не отвечает или ее сетевой интерфейс инициализируется некорректно:

1. Остановите виртуальную машину.
2. Для остановленной виртуальной машины выполните принудительный перезапуск.

Если это не решило проблему, выполните остальные шаги и перейдите к диагностике.

1. Убедитесь, что виртуальная машина запущена.

2. Убедитесь, что вы можете получить доступ к VNC-консоли виртуальной машины и авторизоваться в ней.

Если это необходимо, восстановите пароль для логина.

3. Получите информацию о конфигурации сетевого интерфейса виртуальной машины:
  1. Перейдите в личный кабинет Гиперус.
  2. Выберите проект, где находится нужная виртуальная машина.
  3. Перейдите в раздел **Облачные вычисления → Виртуальные машины**.
  4. Нажмите на имя нужной виртуальной машины.
  5. Перейдите на вкладку **Сети**.
  6. Запишите следующую информацию о сети:
    - имя сети и подсети;
    - шлюз и CIDR для подсети;
    - IP-адреса: приватный IP-адрес и плавающий IP-адрес (если есть);
    - MAC-адрес;
    - настройки Firewall (список групп безопасности).

Для примера будут использоваться следующие значения:

Параметр	Значение
Имя сети	demoNetwork
Имя подсети	demoSubnet
Шлюз	10.0.0.1
CIDR	10.0.0.0/24
Приватный IP-адрес	10.0.0.5
Приватный IP-адрес в комбинации с префиксом из CIDR	10.0.0.5/24
Плавающий IP-адрес	192.0.2.22
MAC-адрес	fa:16:3e:aa:bb: cc
Настройки Firewall	default

## Проверьте настройки сетевого интерфейса

Иногда подключению препятствуют некорректная инициализация сетевого интерфейса виртуальной машины или его неверные настройки.

Проверьте корректность настройки сетевого интерфейса:



1. Подключитесь к консоли виртуальной машины и авторизуйтесь.
2. Выполните команду:

```
1 ip link show
```

Пример вывода:

```
1 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
   UNKNOWN mode DEFAULT group default qlen
   1000
2   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3 2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
   fq_codel state UP mode DEFAULT group
   default qlen 1000
4   link/ether fa:16:3e:aa:bb:cc brd ff:ff:ff:ff:ff:ff
5   altname enp0s3
```

Найдите в выводе команды имя интерфейса, для которого параметр `link\ether` совпадает с MAC-адресом, полученным ранее. В данном примере это будет `ens3`.

Для разных дистрибутивов Linux имя интерфейса будет различаться.

3. Выполните команду, подставив в нее полученное на предыдущем шаге имя интерфейса:

```
1 ip address show ens3
```

Пример вывода:

```
1 2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
   fq_codel state UP group default qlen 1000
2   link/ether fa:16:3e:aa:bb:cc brd ff:ff:ff:ff:ff:ff
3   altname enp0s3
4   inet 10.0.0.5/24 metric 100 brd 10.0.0.255 scope global
   dynamic ens3
5   valid_lft 603373sec preferred_lft 603373sec
6   inet6 fe80::f816:3eff:feb4:d70f/64 scope link
7   valid_lft forever preferred_lft forever
```

Вывод должен содержать:

- Сведения о состоянии интерфейса: `state UP`.
- Приватный IP-адрес виртуальной машины, скомбинированный с префиксом /24 из CIDR подсети, в параметре `inet` (`10.0.0.5/24`).

4. Выполните команду:

```
1 ip route show default
```

Пример вывода:

```
1 default via 10.0.0.1 dev ens3 proto dhcp src 10.0.0.5 metric 100
```

Убедитесь, что вывод содержит:

- IP-адрес шлюза (`via 10.0.0.1`).
- Имя интерфейса, полученное ранее (`dev ens3`).
- Приватный IP-адрес виртуальной машины (`src 10.0.0.5`).

Если вывод команд `ip address show` и `ip route show` содержит в себе приведенные сведения, то настройки сетевого интерфейса корректны. Перейдите к проверке приложений.

Если вывод команд `ip address show` и `ip route show` не содержит в себе приведенных сведений, то настройки сетевого интерфейса некорректны.

## 5. Настройте сетевой интерфейс вручную:

### Debian

1. Отредактируйте файл `/etc/netplan/50-cloud-init.yaml` и приведите его к следующему виду:

```
1 network:
2   ethernets:
3     ens3: # Имяинтерфейса
4           dhcp4: false
5           addresses:
6             - 10.0.0.5/24 # Приватный IPадрес- +
              префиксиз CIDR
7           routes:
8             - to: 0.0.0.0/0
9               via: 10.0.0.1 # Адресшлюза
10          nameservers:
11            addresses:
12              - 5.61.237.120
13              - 5.61.237.127
14          match:
15            macaddress: fa:16:3e:aa:bb:cc # MACадрес-
16            set-name: ens3
17          version: 2
```

При необходимости укажите другие DNS-серверы в параметре `networks. ethernets.ens3.nameservers.addresses`.

2. Выполните команду:

```
1 sudo netplan apply
```

### Ubuntu

1. Отредактируйте файл `/etc/network/interfaces.d/50-cloud-init` и приведите его к следующему виду:

```
1 # This file is generated from information provided by the
   datasources. Changes
```

```
2 # to it will not persist across an instance reboot. To
   # disable cloud-init's
3 # network configuration capabilities, write a file
4 # /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with
   # the following:
5 # network: {config: disabled}
6 auto lo
7 iface lo inet loopback
8
9 auto eth0 # Имяинтерфейса
10 iface eth0 inet static
11 address 10.0.0.5/24 # Приватный IPадрес- + префиксиз CIDR
12 gateway 10.0.0.1 # Адресшлюза
13 dns-nameservers 5.61.237.120 5.61.237.127
```

При необходимости укажите другие DNS-серверы в параметре `dns-nameservers`.

2. Перезапустите сетевое соединение, выполнив команду:

```
1 sudo systemctl restart networking
```

6. Запретите вносить автоматические изменения в отредактированный конфигурационный файл:

```
1 echo 'network: {config: disabled}' | sudo tee /etc/cloud/cloud.
   cfg.d/99-disable-network-config.cfg
```

7. Проверьте наличие доступа по SSH к виртуальной машине. Если доступ не появился, перейдите к проверке приложений.

## Проверьте, что нужные приложения запущены на виртуальной машине

Сетевой интерфейс может быть исправен, но приложения и сервисы могут быть не запущены или работать на нестандартном порте.

Проверьте работу SSH:

1. Подключитесь к консоли виртуальной машины и авторизуйтесь.
2. Выполните команду:

```
1 sudo systemctl status ssh
```

По выводу команды определите, запущен ли сервис:

- `Active: active (running)`: сервис SSH запущен.
- `Active: inactive (dead)`: сервис SSH не запущен.

3. В зависимости от статуса сервиса SSH выполните следующие действия:

*Если сервис SSH запущен*

1. Определите, на каком порте работает сервис SSH:

```
1 sudo cat /etc/ssh/sshd_config | grep -w Port
```

В выводе будет содержаться номер порта:

```
1 Port 22
```

Если сервис работает на стандартном порте 22 — перейдите к проверке настроек фаервола VM. В противном случае переходите к следующему шагу.

2. Подключитесь, используя номер нестандартного порта. Например, если сервис SSH работает на порте 222:

```
1 ssh -i /path/to/private_key_file username@192.0.2.22 -p 222
```

3. Проверьте наличие доступа по SSH к виртуальной машине. Если доступ не появился, перейдите к проверке настроек фаервола VM.

*Если сервис SSH не запущен*

1. Определите, на каком порте работает сервис SSH:

```
1 sudo cat /etc/ssh/sshd_config | grep -w Port
```

В выводе будет содержаться номер порта:

```
1 Port 22
```

2. Убедитесь, что другие процессы не используют этот порт. Просмотрите логи сервиса SSH:

```
1 sudo journalctl -xeu ssh
```

Если в логах есть подобная строка, то порт сервиса SSH используется другим процессом:

```
1 error: Bind to port 22 on 0.0.0.0 failed: Address already in use.
```

Если в логах нет такой строки, перейдите к проверке настроек фаервола VM.

3. Определите, какой процесс занял порт:

1. Установите утилиту `netstat`:

```
1 sudo apt install net-tools -y
```

2. Выполните команду:

```
1 sudo netstat -plntu | grep :22
```

В этом примере вывода порт 22 используется процессом `some-other-service` с PID 1234:

```
1 tcp      0      0 0.0.0.0:22      0.0.0.0:*      LISTEN    1234/some-
   other-service
2 tcp6    0      0 :::22          :::*           LISTEN    1234/some-
   other-service
```

4. Остановите процесс, использующий порт SSH:

- либо с помощью `systemctl`:

```
1 sudo systemctl stop some-other-service
```

- либо принудительно завершив работу процесса:

```
1 sudo kill 1234
```

5. Измените настройки сервиса, соответствующего остановленному процессу, чтобы сервис использовал порт, отличный от 22.

6. Перезапустите сервис:

```
1 sudo systemctl restart some-other-service
```

7. Перезапустите сервис SSH:

```
1 sudo systemctl restart sshd
```

8. Убедитесь, что сервис SSH успешно запущен:

```
1 sudo systemctl status sshd
```

Пример части вывода:

```
1 Active: active (running)
```

Если сервис SSH не запускается, проверьте настройки сервиса в конфигурационном файле `/etc/ssh/sshd_config`. Для получения дополнительной информации о проблемах в работе сервиса посмотрите логи:

```
1 sudo journalctl -xeu ssh
```

9. Проверьте наличие доступа по SSH к виртуальной машине. Если доступ не появился, перейдите к проверке настроек фаервола VM.

## Проверьте настройки фаервола виртуальной машины

Если на виртуальной машине настроен фаервол (например, `iptables`, `ufw`, `firewalld`), то он может препятствовать подключению, даже если IP-адрес виртуальной машины корректный, а сервис SSH настроен и работает.

Далее будет показано, как временно отключить все правила фаервола, разрешив весь трафик. Это поможет убедиться, что проблема именно в фаерволе.

После того, как проблема с SSH-подключением будет локализована, включите правила фаервола обратно (с необходимыми корректировками). Если весь трафик разрешен, безопасность виртуальной машины снизится.

Чтобы проверить настройки фаервола:

1. Подключитесь к консоли виртуальной машины и авторизуйтесь.
2. Отключите фаервол:

```
1 sudo systemctl stop firewalld
```

3. Проверьте наличие доступа по SSH к виртуальной машине.

Если доступ появился, скорректируйте правила фаервола и снова включите его.

Если доступ не появился, снова включите фаервол и проверьте настройки групп безопасности фаервола Гиперус.

Чтобы снова включить фаервол:

1. Подключитесь к консоли виртуальной машины и авторизуйтесь.
2. Выполните команду:

```
1 sudo systemctl start firewalld
```

## Проверьте настройки групп безопасности фаервола Гиперус

Некорректно настроенные группы безопасности могут препятствовать подключению по SSH, даже если на уровне виртуальной машины препятствий нет.

Далее будет показано, как временно настроить правила фаервола так, чтобы разрешить весь трафик. Это поможет убедиться, что проблема именно в фаерволе.

После того, как проблема с SSH-подключением будет локализована, заново настройте правила фаервола (с необходимыми корректировками). Если весь трафик разрешен,

безопасность виртуальной машины снизится.

Чтобы проверить настройки фаервола:

1. Перейдите в личный кабинет Гиперус.
2. Выберите проект, где находится нужная виртуальная машина.
3. Перейдите в раздел **Облачные вычисления → Виртуальные машины**.
4. Нажмите на имя нужной виртуальной машины.
5. Перейдите на вкладку **Сети**.
6. Нажмите “inline” для нужного сетевого подключения и выберите пункт **Редактировать подключение**.
7. В параметре **Настройки Firewall**:
  1. Удалите все выбранные группы безопасности.
  2. Выберите из выпадающего списка группы безопасности **default** и **all** («все разрешено»).

Группа безопасности **default** разрешает любой исходящий трафик. Группа безопасности **all** разрешает любой входящий трафик.

8. Нажмите кнопку **Сохранить**.
9. Проверьте наличие доступа по SSH к виртуальной машине.

Если доступ появился, скорректируйте группы безопасности фаервола и добавьте их снова вместо группы **all**.

Если доступ не появился, вернитесь к исходным настройкам фаервола и обратитесь в техническую поддержку.

Чтобы снова настроить правила фаервола:

1. Перейдите в личный кабинет Гиперус.
2. Выберите проект, где находится нужная виртуальная машина.
3. Перейдите в раздел **Облачные вычисления → Виртуальные машины**.
4. Нажмите на имя нужной виртуальной машины.
5. Перейдите на вкладку **Сети**.
6. Нажмите “inline” для нужного сетевого подключения и выберите пункт **Редактировать подключение**.
7. В параметре **Настройки Firewall**:
  1. Удалите группу безопасности **all**.

2. Выберите из выпадающего списка необходимые группы безопасности.

Если выбранные группы безопасности не содержат правила, разрешающие исходящий трафик, выберите также группу безопасности **default**. Эта группа разрешает исходящий трафик. В противном случае у виртуальной машины не будет доступа в сеть.

Чтобы фаервол пропускал трафик SSH-сервиса, работающего на стандартном порте 22, достаточно выбрать правила **default** и **ssh** («разрешен только ssh»).

## Обратитесь в техническую поддержку

Если диагностика не помогла в решении проблемы, обратитесь в техническую поддержку, предоставив информацию, полученную при диагностике.



# Глава 4. Порядок создания объектов vDC

Все объекты по умолчанию создаются в зоне видимости Проекта и принадлежат ему. Администратор платформы может создавать Проекты и Пользователей, а также присваивать им роли Администраторов в созданных Проектах. Также Администратор может отметить Образы, Конфигурации и Сети как “Общие”, что сделает их видимыми на уровне всей платформы.

## Начало работы

Порядок создания объектов во многом зависит от задачи, которая стоит перед пользователем. Рассмотрим самый простой сценарий: создание новой VM с доступом к интернету и возможностью подключения извне по `ssh`.

## Создание VM

Первым делом выполните вход в портал. Учётная запись администратора по умолчанию которую вам выдаст администратор инсталляции.

## Создание конфигурации

Конфигурация — это шаблон, по которому виртуальной машине будут предоставлены ресурсы. Она включает в себя такие параметры, как количество vCPU и объём оперативной памяти, которыми будет оснащён любой новый VM с этой конфигурацией.

## Создание образа

Образ виртуальной машины — это заранее подготовленный файл, содержащий в себе предустановленную операционную систему.

## Создание сети

**Сеть** Для коммуникации с внешним миром нужно создать сеть — сеть второго уровня модели OSI, обеспечивающая физическую связность. Существует два типа сетей: внутренние и внешние. Для пользователя они отличаются состоянием чекбокса «Внешняя», который доступен при создании сети.

В нашем сценарии нужно создать сети обоих типов.

**Подсеть** Подсеть — это пул IP-адресов и сопутствующих настроек, которые обеспечивают VM возможность взаимодействия с другими сетевыми устройствами на третьем уровне OSI. По умолчанию при создании подсети выбрана опция DHCP, которая позволит виртуальным машинам получать IP-адреса из доступного диапазона автоматически. Подсеть нужно создать для обеих созданных сетей: внутренней и внешней.

**Роутер** Для того чтобы пакеты могли проходить из внутренних сетей во внешний мир и наоборот, нужно создать маршрутизатор — роутер, и подключить к нему созданные сети.

## Непосредственное создание VM

Основные необходимые объекты созданы, и теперь можно приступить непосредственно к созданию виртуальной машины. При создании потребуется указать конфигурацию, образ и сеть, а после того, как VM создастся и запустится, настроить к ней доступ.

Теперь можно проверить доступ к вашему первому VM, подключившись по ssh на IP-адрес, ассоциированный с внутренним адресом VM.

## Создание зон доступности

Зона доступности (AZ) – это логическое разделение инфраструктуры в облачной платформе, обеспечивающее изоляцию ресурсов и повышение отказоустойчивости. AZ не является физически определяемым объектом; это абстракция, скрывающая детали физического расположения оборудования. Ресурсы внутри одной AZ физически отделены от ресурсов других AZ, минимизируя влияние локальных сбоев (отказы оборудования, перебои электропитания и т.д.).

Важно понимать два ключевых аспекта взаимодействия узлов и AZ:

- **Физическое местоположение:** Каждый узел физически находится только в одной AZ. Не существует механизма, позволяющего одному физическому узлу одновременно находиться в нескольких AZ.
- **Логическая многозонность:** Хотя узел физически расположен в одной AZ, программное обеспечение может обеспечивать доступ к нему из других AZ посредством механизмов балансировки нагрузки и распределённых систем. Это позволяет создавать

распределённые приложения, устойчивые к отказам отдельных AZ. Однако, сам узел остаётся в своей исходной AZ. При сбое в одной AZ, доступность приложения может снизиться, но сам узел продолжит функционировать, если его исходная AZ доступна.

Каждый узел при создании назначается в основную зону доступности. Это местоположение по-умолчанию, используемое до тех пор, пока не будут предприняты действия по изменению конфигурации для обеспечения доступа из других AZ.

В целом, AZ – это механизм обеспечения доступности на уровне инфраструктуры, позволяющий проектировать отказоустойчивые приложения путём распределения ресурсов по нескольким физически изолированным зонам. Логическая многозонность достигается программными методами, а не физическим расположением узлов в нескольких AZ одновременно.

## Конфигурации VM

Конфигурация VM представляет собой предопределенный шаблон, описывающий набор аппаратных ресурсов, выделяемых для каждой создаваемой виртуальной машины. Этот шаблон служит основой для процесса виртуализации, обеспечивая предсказуемость и управляемость процесса выделения ресурсов.

Основные параметры конфигурации:

Конфигурация VM определяет набор ключевых параметров, которые должны быть предоставлены физическим узлом для успешного запуска и функционирования виртуальной машины. Эти параметры включают в себя:

- **Количество виртуальных ядер ЦП (vCPU):** Определяет количество логических процессоров, доступных для виртуальной машины. Это значение напрямую влияет на вычислительную мощность VM и её способность обрабатывать параллельные задачи. Выбор количества vCPU должен основываться на предполагаемой рабочей нагрузке. Важно учитывать соотношение между количеством vCPU и количеством доступных физических ядер на целевом узле, чтобы избежать перегрузки системы. Система может ограничивать максимальное количество vCPU, которое может быть запрошено для одной VM, в зависимости от возможностей физической инфраструктуры.
- **Объём оперативной памяти (RAM) в мегабайтах (MB):** Определяет количество оперативной памяти, предоставляемой виртуальной машине. Достаточный объём ОЗУ критически важен для производительности VM. Недостаток памяти может привести к снижению производительности, зависаниям и, в крайних случаях, к краху системы. Аналогично vCPU, максимальный объём RAM для одной VM может быть ограничен возможностями физического узла и общими ресурсами системы виртуализации. Важно учитывать соотношение между запрашиваемым объёмом RAM и доступными ресурсами на физическом узле для предотвращения проблем с перегрузкой памяти.

- **Тип и размер хранилища:** Это критически важный аспект конфигурации. Конфигурация должна определять тип используемого хранилища (например, SSD, HDD), его объём и тип доступа (например, блочное хранилище, объектное хранилище). Выбор типа хранилища оказывает значительное влияние на производительность ввода-вывода VM. Быстрые SSD обеспечивают более высокую производительность для приложений, чувствительных к времени отклика, тогда как HDD подходят для задач с меньшими требованиями к производительности.
- **Сеть:** Конфигурация VM должна определять параметры сетевого подключения, включая тип сети (например, виртуальная локальная сеть - VLAN), IP-адрес, подсеть, шлюз и другие сетевые параметры. Это определяет возможность VM взаимодействовать с другими системами в сети.
- **Устройства виртуальных дисков:** Здесь указываются виртуальные диски, которые будут подключены к VM. Это включает в себя размер, тип и формат диска.

### **Управление доступом к конфигурациям:**

Система управления конфигурациями VM обеспечивает гибкий механизм контроля доступа, позволяя администраторам платформы регулировать, какие проекты могут использовать определенные конфигурации. Это обеспечивает безопасность и предотвращает несанкционированное использование ресурсов.

- *Общедоступные конфигурации:* Эти конфигурации доступны для всех проектов на платформе. Их использование упрощает создание VM, но требует тщательного контроля над ресурсами, чтобы предотвратить перегрузку системы.
- *Частные конфигурации:* Эти конфигурации доступны только для определённых проектов или доменов. Это позволяет администраторам ограничивать использование ресурсов, настраивать выделенные конфигурации для специфических задач или приложений и обеспечивает более тонкий контроль над распределением ресурсов. Доступ к частным конфигурациям может быть основан на различных политиках безопасности и управляется средствами контроля доступа.
- *Управление квотами:* В дополнение к управлению доступом, система должна поддерживать управление квотами на ресурсы. Это позволяет ограничивать общее количество ресурсов, которые могут быть запрошены проектом, предотвращая злоупотребления и обеспечивая справедливое распределение ресурсов среди всех пользователей. Квоты могут быть установлены на уровне vCPU, RAM, хранилища и других ресурсов.
- *Управление жизненным циклом конфигураций:* Администраторы имеют возможность создавать, редактировать, удалять и управлять версиями конфигураций VM. Это позволяет адаптироваться к изменениям потребностей и обновлять параметры конфигурации по мере необходимости.

### **Процесс выбора конфигурации:**

При создании VM пользователь выбирает одну из доступных конфигураций, руководствуясь требованиями своего приложения. Система автоматически проверяет, достаточно ли ресурсов на физических узлах для удовлетворения запроса, и, в случае успешной проверки, начинает процесс создания VM с указанной конфигурацией. В случае нехватки ресурсов, система может отказать в создании VM или предложить альтернативные варианты.

### **Взаимодействие с другими компонентами системы:**

Конфигурации VM тесно взаимодействуют с другими компонентами платформы, такими как планировщик (scheduler), драйверы гипервизоров и системы мониторинга. Планировщик использует информацию о конфигурациях для выбора подходящего физического узла для размещения VM. Драйверы гипервизоров отвечают за фактическое создание и управление VM на физическом узле в соответствии с указанной конфигурацией. Системы мониторинга отслеживают использование ресурсов VM и сообщают о любых аномалиях.

Администратор платформы указывает проекты, которые могут использовать Конфигурацию, сделав ее общедоступной для всех проектов или частной для определенных проектов или доменов.

## **Создание конфигураций**

Вы можете создавать и управлять различными вариантами Конфигураций для достижения определенных функций или поведения, например:

- Указывать количество виртуальных ЦП;
- Указывать количество оперативной памяти и емкость основного диска по умолчанию в соответствии с потребностями VM;
- Добавлять метаданные, чтобы установить определенную скорость ввода-вывода для VM или ограничивать запуск VM на определенном агрегате узлов.

Администратор платформы может создать необходимое количество Конфигураций. Конфигурации создаются как через портал, так и через CLI. Конфигурация создается в подразделе

**Конфигурации VM** раздела меню **Конфигурация** портала следующим образом:

- нажать кнопку Создать на панели инструментов;
- заполнить открывшуюся форму Создание конфигурации виртуальных машин;
- нажать кнопку Создать.

## **Использование CPU-pinning**

В ряде задач, таких как обработка в реальном времени (Real-Time) или близком к реальному времени (Near Real-Time), критично минимизировать задержки, вносимые гипервизором и планировщиком операционной системы. Стандартные методы распределения ресурсов

ЦП, используемые по умолчанию в большинстве виртуализационных платформ, могут вводить непредсказуемые задержки, делая невозможной корректную работу подобных приложений. Для решения этой проблемы применяется техника CPU-pinning (закрепление ЦП).

CPU-pinning – это механизм, позволяющий привязать виртуальные ЦП (vCPU) виртуальной машины (ВМ) к конкретным физическим ядрам процессора (PCPU) узла. Это позволяет эффективно использовать выделенные вычислительные ресурсы, минимизируя конкуренцию за процессорное время и, следовательно, снижая задержки.

### **Ключевые аспекты CPU-pinning:**

- **Изоляция ресурсов:** ВМ с закреплёнными ЦП (CPU-pinned VMs) получают гарантированный доступ к определённому набору физических ядер. Они не могут использовать ЦП, назначенные другим ВМ с включённым CPU-pinning, что исключает конфликты за ресурсы и обеспечивает предсказуемое поведение.
- **Различие в управлении ресурсами:** Гипервизор обрабатывает ресурсы процессора по-разному для ВМ с закреплёнными и незакреплёнными ЦП.
- **ВМ без CPU-pinning:** Используют тип ресурса vCPU. Для этих ВМ допустима переподписка ресурсов ЦП (oversubscription), то есть количество vCPU может превышать количество доступных PCPU на узле. Это достигается за счёт динамического распределения ресурсов процессора между несколькими ВМ.
- **ВМ с CPU-pinning:** Используют тип ресурса PCPU. Для этих ВМ переподписка ресурсов недопустима. Каждому vCPU ВМ однозначно сопоставляется один PCPU. Это гарантирует выделение ресурсов и предсказуемость работы.
- **Поведение по умолчанию:** Без специальной настройки, Гиперус.Инфраструктура по умолчанию рассматривает все ЦП узла как vCPU, позволяя переподписку ресурсов.
- **Настройка параметров:** Поведение системы по умолчанию можно изменить, используя параметры конфигурации. Эти параметры позволяют настраивать механизмы распределения ресурсов, включая управление CPU-pinning.

### **Преимущества использования CPU-pinning:**

- **Минимальные задержки:** Гарантированное выделение физических ядер процессора минимизирует задержки, что критично для приложений, чувствительных к задержкам.
- **Предсказуемость производительности:** Использование выделенных ресурсов делает производительность ВМ более предсказуемой и стабильной.
- **Изоляция:** Предотвращение конфликтов за ресурсы между ВМ с закреплёнными ЦП повышает надёжность и стабильность работы системы в целом.

### **Недостатки использования CPU-pinning:**

- **Меньшая эффективность использования ресурсов:** CPU-pinning может привести к снижению эффективности использования ресурсов ЦП, так как часть ядер может оставаться неиспользуемой, если ВМ не полностью задействует выделенные ресурсы.

- **Усложнение управления ресурсами:** Управление ресурсами становится более сложным, так как требует ручного или автоматизированного назначения физических ядер процессора для каждой VM.
- **Ограничения на масштабируемость:** Внедрение CPU-pinning может ограничить масштабируемость системы, так как каждый PCPU может быть использован только одной VM.

CPU-pinning – мощный инструмент для оптимизации производительности приложений, требующих минимальных задержек. Однако, его использование требует тщательного планирования и понимания компромисса между производительностью и эффективностью использования ресурсов. Необходимо тщательно оценить преимущества и недостатки перед включением CPU-pinning для конкретных VM. Правильная настройка параметров конфигурации является ключевым фактором для успешного внедрения CPU-pinning.

## Использование Квот и QoS

С помощью метаданных Конфигураций вида quota: возможно определять различные квоты для VM, например, квоты CPU-времени выполнения на гипервизоре для данной VM. Все VM по-умолчанию создаются с квотой выполнения 1024. Если в метаданные передать `quota:cpu_shares = 512`, то все VM с этим профилем будут получать в два раза меньше процессорного времени на гипервизоре (аналог `cpu_shares` для `systemd`) при конкуренции с другими VM за ресурсы ЦП. Также может быть полезно включить механизм сетевого шейпинга (QoS) для интерфейсов VM через метаданные.

## Планировщик VM

Гиперус.Инфраструктура использует службу планирования, чтобы определить, на каком физическом узле или агрегате узлов следует разместить экземпляр VM. Когда служба получает запрос на запуск или миграцию VM, она использует указанные в запросе спецификации, конфигурацию и образ для поиска подходящего узла. Например, в конфигурации с помощью метаданных можно указывать различные признаки, по которым будет выбран узел для запуска экземпляра VM: тип диска, расширение набора инструкций ЦП, выделенный GPU и многое другое. Служба запускается на всех узлах с ролью “Управление VM”. В контексте фильтров термин «узел» означает физический узел с ролью “Физический узел”, на котором запущена служба. Служба планировщика для определения узла, на котором нужно запустить или переместить VM, использует компоненты в следующем порядке :

1. **Предварительные фильтры.** Служба планировщика вычислений использует набор узлов-кандидатов на основе определенных атрибутов.
2. **Фильтры.** Используются службой планировщика для определения начального набора физических узлов, на которых следует запускать экземпляр VM.

3. **Вес.** Служба планировщика отдает приоритет отфильтрованным физическим узлам с помощью системы взвешивания. Наибольший вес имеет наивысший приоритет.

## Фильтрация по характеристикам поставщика ресурсов

У каждого поставщика ресурсов есть набор специальных характеристик. Это качественные аспекты поставщика ресурсов, например, тип диска хранения или расширение набора инструкций ЦП.

Физический узел сообщает о своих возможностях в виде специальных характеристик. Таким образом, экземпляр VM может указать, какие из этих характеристик ему необходимы, а какие не должны присутствовать у поставщика ресурсов. Планировщик физических ресурсов может использовать эти признаки для определения подходящего физического узла или агрегата узлов для размещения экземпляра VM. Чтобы позволить пользователям создавать VM на узлах с определенными характеристиками, можно определить вариант Конфигурации, требующий или запрещающий определенную характеристику. Также можно создать образ, требующий или запрещающий ту или иную характеристику.

## Вес

Взвешиваются только узлы, прошедшие все фильтры. Вес любого узла начинается с 0. Взвешивание упорядочивает эти узлы, добавляя или вычитая из значения веса узла, назначенного предыдущим взвешивающим. Веса могут быть отрицательными. Экземпляр VM будет назначен на один из узлов с наибольшим весом. Планировщик определяет вес каждого физического узла, выполняя следующие задачи:

- Планировщик нормализует каждый вес до значения от 0,0 до 1,0;
- Планировщик умножает нормализованный вес на множитель веса.

Планировщик вычисляет весовую нормализацию для каждого типа ресурсов, используя нижнее и верхнее значения доступности ресурсов для физических узлов-кандидатов:

- узлам с наименьшей доступностью ресурса (`minval`) присваивается 0;
- узлам с наивысшей доступностью ресурса (`maxval`) присваивается 1;
- узлам с доступностью ресурсов в диапазоне `minval-maxval` назначается нормализованный вес, рассчитываемый по следующей формуле:  $(\text{node\_resource\_availability} - \text{minval}) / (\text{maxval} - \text{minval})$ .

Если все физические узлы имеют одинаковую доступность ресурса, то все они нормализуются до 0. Планировщик фильтров взвешивает узлы на основе параметра конфигурации.



## Диски и СХД

Данный раздел описывает архитектуру и механизмы работы службы блочного хранилища Гиперус, включая управление дисками виртуальных машин (ВМ), типы бэкендов и методики безопасного удаления данных.

### Бэкенды

Служба блочного хранилища Гиперус.Инфраструктура отвечает за управление жизненным циклом дисков, используемых виртуальными машинами. Она запускается на выделенных узлах с ролью “Управление дисками” (или аналогичной) и обеспечивает администрирование, безопасность, планирование и мониторинг всех дисков, используемых экземплярами ВМ. Диски служат основной формой постоянного хранилища для данных ВМ.

### Описание работы компонентов

Служба блочного хранилища Гиперус.Инфраструктура устанавливается на узлы с ролью “Управление дисками” и управляет администрированием, безопасностью, планированием и жизненным циклом всех дисков ВМ. Диски используются в качестве основной формы постоянного хранилища для экземпляров ВМ. Метод предоставления и использования хранилища определяется драйвером хранилища.

### Безопасное удаление дисков ВМ

Для соответствия требованиям информационной безопасности, Гиперус.Инфраструктура предоставляет несколько методик удаления дисков виртуальных машин. Выбор метода определяется параметром `volume_clear` в конфигурации бэкенда хранилища. Этот параметр может принимать следующие значения:

- **none:** Файл диска удаляется с помощью стандартной системной команды `rm`. Этот метод быстрый, но не гарантирует полное удаление данных. Остаточные данные могут быть восстановлены с помощью специальных инструментов. Этот метод не рекомендуется для сред с высокими требованиями к безопасности.
- **zero:** Файл диска перезаписывается нулями в один проход с использованием утилиты `dd` и затем удаляется командой `rm`. Эта методика более надёжна, чем `none`, так как перезапись нулями затрудняет восстановление данных. Однако, данный метод значительно увеличивает время удаления диска, пропорционально его размеру. Полный размер диска перезаписывается нулями, что может занять значительное время для больших дисков.
- **shred:** Используется утилита `shred` для многократной перезаписи диска случайными данными, что делает восстановление данных практически невозможным. Это самый надёжный, но и самый медленный метод.

**Критические соображения:**

Выбор метода удаления дисков должен учитывать баланс между безопасностью и производительностью. В средах с высокими требованиями к безопасности рекомендуется использовать zero или shred, осознавая потенциальное увеличение времени удаления. Для сред с меньшими требованиями к безопасности может быть достаточно метода wipe, но следует помнить о риске восстановления данных. Важно документировать выбранный метод и учитывать его влияние на производительность системы. Важно также учитывать, что даже метод zero или shred не гарантирует 100% удаления данных, особенно в случае использования специализированных инструментов для восстановления данных.

## Глава 5. Сеть

Сетевая подсистема ПО Гиперус.Инфраструктура играет ключевую роль в обеспечении связи между виртуальными машинами, а также между ними и внешними ресурсами. Все компоненты сетевой подсистемы настраиваются и конфигурируются в процессе развертывания платформы. В данной главе будет рассмотрено, как правильно настраивать и управлять сетевыми компонентами.

### Настройка основных компонентов

1. Роль узла - Для установки основного компонента сетевой подсистемы необходимо активировать роль «Управление сетями» в параметрах узла. Эта роль включается автоматически при выборе профилей «Основной» или «Дополнительный» при добавлении нового узла в систему.
2. Установка дополнительных компонентов - Дополнительные компоненты сетевой подсистемы устанавливаются при активации роли «Физический узел». Данная роль назначается при использовании профилей «Арбитр» и «Вычислительный узел». Обратите внимание, что каждая роль определяет, какие компоненты и функции будут доступны на конкретном узле.

### Необходимые ресурсы для функционирования сетевых служб

Для эффективного функционирования сетевых служб в системе необходим доступ к следующим ресурсам:

- Экземпляр базы данных - Сетевые службы должны иметь доступ к экземпляру базы данных. Это может быть один узел или несколько узлов, работающие в режиме высокой доступности (HA). Это обеспечивает отказоустойчивость и доступность данных, необходимых для работы сетевых функций.
- Работающая служба сообщений - Услуга NATS должна быть развернута и доступна для всех компонентов сетевой подсистемы. NATS используется для обмена сообщениями между различными службами, обеспечивая надежную и быструю передачу данных.

- Служба кэширования Redis - Доступ к экземпляру Redis необходим для кэширования данных и ускорения работы сетевых служб. Redis обеспечивает быстрый доступ к часто используемым данным, что улучшает общую производительность системы.
- Служба идентификации - Все сетевые компоненты должны иметь доступ к службе идентификации (например, Keystone) для аутентификации и авторизации пользователей и служб. Это гарантирует безопасность и контроль доступа к ресурсам облака.

# Глава 6. Кластеры

## Введение

Кластер представляет собой совокупность хостов, которые можно рассматривать как независимый объект в системе, обладающий расширенными свойствами и функционалом. Логическое объединение ресурсов всех подключённых хостов формирует единую ресурсную среду, что позволяет выполнять задачи, требующие больше ресурсов, чем способен предоставить отдельный хост. Объединение хостов в кластер открывает возможности для реализации механизма отказоустойчивости, что обеспечивает стабильную работу пользователей с виртуальными инфраструктурами даже в случае сбоев оборудования.

В кластере возможно применение функции распределения нагрузки между хостами, использующей алгоритмы автоматического балансирования. В системе Гиперус.Инфраструктура за эту задачу отвечает ресурсный планировщик, который регулярно оценивает загрузку хостов и перераспределяет ресурсы для запуска и поддержки виртуальных сред с более загруженных хостов на менее нагруженные.

Планировщик активируется при возникновении следующих событий:

- **Запуск виртуальной среды** — происходит пересчёт ресурсов как хоста, так и кластера, что позволяет принять решение о размещении и активации виртуальной среды;
- **Остановка виртуальной среды** — осуществляется пересчёт ресурсов хоста и всего кластера;
- **Миграция виртуальной среды** — проводится выбор целевого хоста и пересчёт его ресурсов;
- **Изменение параметров работающей виртуальной среды** — пересчитываются ресурсы хоста и кластера, принимается решение о необходимом изменении ресурсов;
- **Запуск хоста** — выполняется пересчёт ресурсов всего кластера;
- **Перевод хоста в режим обслуживания** — осуществляется пересчёт ресурсов кластера и перемещение работающих виртуальных сред на оставшиеся узлы кластера;
- **Изменение параметров работающего хоста** — производится пересчёт ресурсов кластера.

В системе Гиперус.Инфраструктура не поддерживается работа одиночных хостов; любой хост должен быть в составе кластера. Администратор может создать кластеры следующих типов:

- Отказоустойчивый кластер (или HA-кластер — high availability, высокодоступный) — кластер с механизмом отказоустойчивости;
- Обычный кластер (non-HA) — логическая группа хостов. В таком кластере нет возможности использовать механизм отказоустойчивости.

# Глава 7. Подготовка VM с ОС Astra Linux

Глава содержит порядок подготовки сервера с ОС Astra Linux Special Edition для установки платформы и настройки узлов кластера.

## Установка ОС

При установке операционной системы:

1. На шаге Установка базовой системы выберите Ядро для установки — `linux-5.15-generic` или `linux-6.1-generic`.
2. На шаге Выбор программного обеспечения:
  - Выберите следующие компоненты:
    - Средства Виртуализации.
    - Консольные утилиты.
    - Средства удаленного подключения SSH.
  - Убедитесь, что остальные компоненты отключены.
3. На шаге Дополнительные настройки ОС:
  - Выберите уровень защищенности “Орел” или “Воронеж”.
  - Убедитесь, что отключены опции:
    - Замкнутая программная среда.
    - Запрет установки бита исполнения.
    - Запрет исполнения скриптов пользователя.

Если вы выбрали уровень защищённости “Воронеж”, рекомендуем отключить опцию “Очистка освобождаемой внешней памяти”. Эта опция значительно увеличивает нагрузку на CPU и время выполнения некоторых операций.

## Настройка ОС на сервере с платформой

(1) Перейдите в режим суперпользователя:

```
1 sudo su -
```

(2) Настройте список источников в файле `/etc/apt/sources.list`:

- для установки в закрытом контуре укажите только источник `cdrom`. Остальные источники должны быть удалены или закомментированы:

Пример файла для Astra Linux Special Edition 1.7.5

```
1 deb cdrom: [OS Astra Linux 1.7.5 1.7_x86-64 DVD]/ 1.7_x86-64
   contrib main non-free
```

- для установки в открытом контуре укажите только источники из поддрева `Frozen` для нужной версии ОС. Остальные источники должны быть удалены или закомментированы:

Пример файла для Astra Linux Special Edition 1.7.4 base (1.7.4-24.04.2023\_14.23.iso)

```
1 deb <http://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.4/repository-main/> 1.7_x86-64 main contrib
   non-free
2 deb <http://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.4/repository-update/> 1.7_x86-64 main contrib
   non-free
3 deb <http://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.4/repository-base/> 1.7_x86-64 main contrib
   non-free
4 deb <http://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.4/repository-extended/> 1.7_x86-64 main
   contrib non-free
```

Пример файла для Astra Linux Special Edition 1.7.4 uu1 (installation-1.7.4.11-23.06.23\_17.13.iso)

```
1 deb <https://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.4/uu/last/repository-base/> 1.7_x86-64 main
   contrib non-free
2 deb <https://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.4/uu/last/repository-extended/> 1.7_x86-64
   main contrib non-free
3 deb <https://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.4/uu/last/repository-main/> 1.7_x86-64 main
   contrib non-free
4 deb <https://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.4/uu/last/repository-update/> 1.7_x86-64 main
   contrib non-free
```

Пример файла для Astra Linux Special Edition 1.7.5

```
1 deb <https://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.5/repository-base/> 1.7_x86-64 main contrib
   non-free
```



```
2 deb <https://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.5/repository-extended/> 1.7_x86-64 main
   contrib non-free
3 deb <https://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.5/repository-main/> 1.7_x86-64 main contrib
   non-free
4 deb <https://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.5/repository-update/> 1.7_x86-64 main contrib
   non-free
5 deb <https://dl.astralinux.ru/astra/frozen/1.7_x86
   -64/1.7.5/uu/last/repository-update/> 1.7_x86-64 main
   contrib non-free
```

- (3) В файле `/etc/network/interfaces` настройте сетевую конфигурацию со статическим IP-адресом:

Пример конфигурации

```
1 auto lo
2 iface lo inet loopback
3
4 auto eth0
5 iface eth0 inet static
6     address    192.168.1.10
7     netmask    255.255.255.0
8     gateway    192.168.1.254
```

- (4) Если платформа будет использоваться в открытом информационном контуре (с доступом к Интернет), создайте файл `/etc/resolv.conf` и укажите в нём настройки DNS-серверов:

Пример настройки

```
1 nameserver 77.88.8.8
2 nameserver 1.1.1.1
3 options timeout:1
4 options attempts:2
```

- (5) Перезапустите службу `networking`:

```
1 sudo systemctl restart networking
```

- (6) Проверьте доступность сети.

## Настройка системного времени

### Открытый контур

На сервере платформы и узлах кластера настройте синхронизацию системного времени с NTP-сервером. Подробнее см. в [документации Astra Linux](#).

## Закрытый контур

Настройте NTP-сервер на сервере с платформой и NTP-клиенты на узлах кластера.

### Настройка NTP-сервера

1. Установите пакеты ПО NTP:

```
1 sudo apt-get update
2 sudo apt-get install ntp
```

2. В конфигурационном файле `/etc/ntp.conf`:

- Отключите синхронизацию с внешними NTP-серверами. Для этого закомментируйте все строки, начинающиеся с `pool` или `server`.
- Подключите локальный NTP-сервер. Для этого добавьте строку:

```
1 server 127.127.1.0 fudge 127.127.1.0 stratum 8
```

3. Перезапустите службу `ntp`:

```
1 sudo /etc/init.d/ntp restart
```

## Настройка сети на узлах кластера

В файле `/etc/network/interfaces` настройте сетевую конфигурацию. Примеры конфигураций:

### Минимальная конфигурация без бриджей и бондов

```
1 auto lo
2 iface lo inet loopback
3
4 auto eth0
5 iface eth0 inet static
6     address    192.168.1.10
7     netmask    255.255.255.0
8     gateway    192.168.1.254
```

### Конфигурация с бриджем и интерфейсом во VLAN

```
1 auto lo
2 iface lo inet loopback
3
4 auto eth0 vubr0 vubr1
```

```
5  iface eth0 inet manual
6
7  auto eth0.111
8  iface eth0.111 inet manual
9      vlan_raw_device eth0
10
11 auto eth0.112
12 iface eth0.112 inet manual
13     vlan_raw_device eth0
14
15 iface vobr0 inet static
16     address 1.1.111.10
17     netmask 255.255.255.0
18     gateway 1.1.111.1
19     bridge_ports eth0.111
20     bridge_stp off
21
22 iface vobr1 inet static
23     address 1.1.112.10
24     netmask 255.255.255.0
25     gateway 1.1.112.1
26     bridge_ports eth0.112
27     bridge_stp off
```

## Конфигурация с бондом во VLAN в режиме active-backup

```
1  auto lo
2  iface lo inet loopback
3
4  auto eth0 eth1 bond0 bond0.118 vobr0
5  iface eth0 inet manual
6  iface eth1 inet manual
7
8  iface bond0 inet manual
9  pre-up modprobe bonding mode=active-backup
10 post-down rmmod bonding
11 bond-mode active-backup
12 bond-primary eth0
13 bond-miimon 100
14 bond-downdelay 200
15 bond-updelay 150
16 bond-slaves eth0 eth1
17
18 iface bond0.118 inet manual
19     vlan-raw-device bond0
20
21 iface vobr0 inet static
22     address 172.22.18.74
23     netmask 255.255.255.0
24     gateway 172.22.18.1
25     bridge_ports bond0.118
```

```
26 bridge_stp off
```

## Конфигурация в режиме **balance-alb** с VLAN

```
1 auto lo
2 iface lo inet loopback
3
4 auto eth2 eth3 bond0 bond0.772 vbr0
5
6 iface eth2 inet manual
7 iface eth3 inet manual
8
9 iface bond0 inet manual
10     bond-mode 6
11     bond-miimon 100
12     bond-slaves eth2 eth3
13
14 iface bond0.772 inet manual
15     vlan-raw-device bond0
16
17 iface vbr0 inet static
18     address 10.77.2.20/24
19     gateway 10.77.2.1
20     bridge_ports bond0.772
21     bridge_stp off
```

## Конфигурация с бондом и бриджем в режиме **LACP (802.3ad)**

```
1 auto lo
2 iface lo inet loopback
3
4 auto eth0 eth1 bond0 vbr0
5 iface eth0 inet manual
6 iface eth1 inet manual
7
8 iface bond0 inet manual
9     bond-mode 802.3ad
10    bond-miimon 100
11    bond-downdelay 200
12    bond-updelay 200
13    bond-xmit-hash-policy 1
14    bond-slaves eth0 eth1
15
16 iface vbr0 inet static
17     address 172.22.18.74
18     netmask 255.255.255.0
19     gateway 172.22.18.1
20     bridge_ports bond0
21     bridge_stp off
```

Перезапустите службу networking:

```
1 sudo systemctl restart networking
```

## Настройка подключения к узлам кластера

Раздел содержит инструкции, которые необходимо выполнить, чтобы платформа могла корректно подключаться к узлам кластера. Авторизация платформы на узлах кластера осуществляется по SSH-ключу.

Корректное подключение узла гарантируется только при использовании учётной записи, отличной от root.

### Уровень защищённости “Орёл”

1. Добавьте публичный SSH-ключ сервера с платформой в файл `/home/[username]/.ssh/authorized_keys` где `[username]` — имя пользователя, под которым платформа будет подключаться к серверу. Содержимое публичного ключа можно посмотреть в интерфейсе платформы на форме добавления узла кластера: раздел Узлы → кнопка Подключить узел → кнопка Хочу использовать публичный ssh-ключ.
2. Установите права для файла `/home/[username]/.ssh/authorized_keys`:

```
1 sudo chmod 600 /home/<username>/.ssh/authorized_keys
2 sudo chown <username> /home/<username>/.ssh/authorized_keys
```

3. Проверьте права доступа на файл `/home/[username]/.ssh/authorized_keys`. Файл должен быть доступен на чтение и запись для пользователя, под которым платформа будет подключаться к серверу.
4. В файле `/etc/sudoers` разрешите выполнение команд от имени суперпользователя:

```
1 %astra-admin ALL=(ALL:ALL) NOPASSWD: ALL
2 %sudo ALL=(ALL:ALL) NOPASSWD: ALL
```

### Уровень защищённости “Воронеж”

1. Убедитесь, что службы `astra-sudo-control`, `astra-nochmodx-lock`, `astra-interpretors-lock` отключены:

```
1 sudo astra-sudo-control disable
2 sudo astra-nochmodx-lock disable
3 sudo astra-interpretors-lock disable
```

2. Создайте директорию для хранения VM (например, /vm) и установите для неё [мандатные атрибуты](#):

```
1 sudo mkdir /vm && sudo pdpl-file 0:63:0:ccnr /vm
```

3. Добавьте в файл /home/[username]/.ssh/authorized\_keys публичный SSH-ключ сервера с платформой. Содержимое публичного ключа можно посмотреть в интерфейсе платформы на форме добавления узла кластера: Узлы → Подключить узел → кнопка Хочу использовать публичный ssh-ключ .
4. Установите права для файла /home/[username]/.ssh/authorized\_keys:

```
1 sudo chmod 600 /home/<username>/.ssh/authorized_keys
2 sudo chown <username> /home/<username>/.ssh/authorized_keys
```

5. Проверьте права доступа на файл /home/[username]/.ssh/authorized\_keys. Файл должен быть доступен на чтение и запись для пользователя, под которым платформа будет подключаться к серверу.
6. В файле /etc/sudoers разрешите выполнение команд от имени суперпользователя:

```
1 %astra-admin ALL=(ALL:ALL) NOPASSWD: ALL
2 %sudo ALL=(ALL:ALL) NOPASSWD: ALL
```

# Решение типовых проблем пользователей ПО (FAQ)

## Как исправить проблему с высокой нагрузкой на кластер, приводящей к медленной работе виртуальных машин?

### Ответ:

1. Проверка загрузки ресурсов: Войдите в интерфейс управления Гиперус.Инфраструктура и перейдите на дашборд. Здесь вы сможете увидеть загрузку CPU, памяти и дискового пространства. Обратите внимание на любые узкие места.
2. Выравнивание нагрузки: Если один из узлов перегружен, вы можете переместить виртуальные машины на другие узлы. Для этого выберите виртуальную машину, которую хотите переместить, и воспользуйтесь функцией “Migrate VM” в меню действий (Actions).
3. Оптимизация конфигурации VM: Проверьте, достаточно ли ресурсов назначено виртуальным машинам. Возможно, их необходимо пересмотреть и увеличить количество выделенных ресурсов (CPU, RAM).
4. Обновление ПО: Убедитесь, что ваша версия Гиперус.Инфраструктура обновлена до последней. Устаревшее ПО может содержать баги, которые влияют на производительность.
5. Контроль за приложениями: Если проблема сохраняется, проверьте, какие приложения запущены на виртуальных машинах. Возможно, некоторые из них требуют слишком много ресурсов, и их функциональность следует оптимизировать.

## Как исправить ошибку “No valid host found” при развертывании виртуальной машины?

### Ответ:

1. Проверка ресурсов: Убедитесь, что есть доступные ресурсы для создания новой виртуальной машины. Проверьте, достаточно ли свободных CPU, памяти и дискового пространства. .

2. Квоты проекта: Убедитесь, что в проекте не достигнут лимит квот на виртуальные машины. При необходимости увеличьте квоты с помощью команды `openstack quota set`.
3. Фильтры и характеристики: Проверьте, что заданные характеристики виртуальной машины (flavor) соответствуют доступным ресурсам на гипервизорах. Если параметр `flavor` требует больше ресурсов, чем доступно на имеющихся узлах, то следует создать VM с меньшими параметрами.
4. Сетевая конфигурация: Убедитесь, что для создания виртуальной машины выбрана правильная сеть, и все настройки корректны. Проверьте, доступны ли ваши сети и подсети, используя команду `openstack network list`.
5. Логи и ошибки: Проверьте логи сервисов для получения более подробной информации об ошибке. Это можно сделать, зайдя в панель управления.



# Термины и сокращения

Термин	Описание
Виртуальная машина (ВМ)	Виртуальная машина внутри проекта
Центр обработки данных (ЦОД)	Выделенное помещение в котором компания размещает серверное и сетевое оборудование
Внешняя сеть	Конфигурация для подключения ВИ к внешней физической сети
НС (health check)	Оценка состояния системы, системных ресурсов
LB (Load Balancer)	Балансировщик нагрузки, предназначенный для распределения сетевого трафика между виртуальными машинами
Snapshot	Моментальный снимок ВМ
Зона доступности	Один или несколько центров обработки данных, в которых могут быть размещены компоненты облачной инфраструктуры.
Пул ресурсов	Группа ресурсов (например, ЦП, память, хранилище), выделяемых для использования виртуальными машинами.
Образ (Image)	Файл, содержащий операционную систему и преднастройки для создания виртуальной машины.
Кластер	Группа серверов, работающих совместно для обеспечения высокой доступности и отказоустойчивости.
Нагрузочный тест	Процесс проверки производительности системы под нагрузкой для выявления узких мест.
Квота	Ограничение на количество ресурсов (например, ВМ, дисков), которые могут быть выделены пользователю или проекту.
Идентификатор проекта	Уникальный идентификатор, использующийся для разделения ресурсов между разными проектами в Гиперус.Инфраструктура

<b>Термин</b>	<b>Описание</b>
API (Application Programming Interface)	Интерфейс для программного взаимодействия с компонентами Гиперус.Инфраструктура
Пользовательская роль	Определённые разрешения для выполнения задач в Гиперус, основанные на ролях пользователей.
Служба мониторинга	Инструмент для отслеживания состояния ресурсов и производительности систем в Гиперус.Инфраструктура
Событие (Event)	Запись о произошедшем событии в системе, используемая для аудита и мониторинга.
Система безопасного удаления	Механизмы, обеспечивающие безопасное и полное удаление данных из хранилищ.
ID объемов	Уникальный идентификатор, используемый для управления блочными хранилищами в Cinder.
Миграция VM	Процесс перемещения виртуальной машины с одного физического узла на другой.
Автоскэйлинг	Процесс автоматического добавления или удаления ресурсов в зависимости от нагрузки.
Теневое хранилище	Хранилище, которое предоставляет доступ к данным, но может быть отключено от основной системы.
Регион	Географически обособленная группа зон доступности в Гиперус.Инфраструктура
Флейвор	Набор ресурсов, описывающий характеристики виртуальной машины (например, количество ЦП и RAM).
Диск (Volume)	Блочное хранилище, выделенное для использования одной или нескольких виртуальных машин.
Персональный облако	Облачная инфраструктура, предоставляемая для использования одного пользователя или группы пользователей.
Гибридное облако	Облачная среда, состоящая из публичного и частного облака, работающих совместно.
Гиперус.Инфраструктура Community	Сообщество разработчиков, которое занимается разработкой и поддержкой Гиперус.Инфраструктура
Защита от DDoS	Механизмы и системные настройки, направленные на защиту от распределённых атак на отказ в обслуживании.
Поддержка VPC (Virtual Private Cloud)	Функционал Гиперус, позволяющий создавать виртуально изолированные сети в облачной инфраструктуре.

<b>Термин</b>	<b>Описание</b>
Heterogeneous Environment	Среда с различными типами и моделями эксплуатации оборудования.
Network Policy	Набор правил, регламентирующих, как сетевые ресурсы могут взаимодействовать друг с другом.
API Rate Limiting	Ограничение частоты использования API для обеспечения доступности и производительности.
Инфраструктура как код (IaC)	Практика управления и прототипирования вычислительных ресурсов с использованием конфигурационных файлов.
Cloud Controller	Управляющий компонент Гиперус, координирующий работу различных сервисов.
SFC (Service Function Chain)	Концепция последовательности сетевых сервисов, упорядоченных для обработки трафика.
Тактическое резервирование	Способ создания резервных копий для быстрого восстановления системы.
Сервер конфигураций	Узел, который управляет настройками и конфигурациями остальных узлов в облаке.
RT (Real-Time) Monitoring	Непрерывный мониторинг состояния системы с возможностью мгновенного реагирования на события.
Provisioning	Процесс создания и настройки ресурсов в облачной среде.
DevOps	Практика совместной работы разработчиков и операционных команд для повышения эффективности.
EOL (End of Life)	Конечный срок, когда продукт или его версия больше не поддерживаются.
Multi-Tenancy	Возможность использования одних и тех же ресурсов несколькими пользователями или проектами без взаимного влияния.
Migration Strategy	План, который описывает подход и действия для перемещения сервисов в облако.
Панель Управление Гиперус.Инфраструктура	Графический интерфейс для отображения показателей состояния систем и сервисов.
End-to-End Encryption	Процесс шифрования данных с момента их создания до момента завершения обработки.
On-Premises	Системы и решения, размещенные на локальном оборудовании организации.

<b>Термин</b>	<b>Описание</b>
Maintenance Window	Запланированный период, в течение которого проводятся работы по обслуживанию систем и приложений.
CPU	(англ. central processing unit — центральное обрабатывающее устройство) — Центральный процессор, электронный блок либо интегральная схема, исполняющая машинные инструкции
DMZ	(англ. Demilitarized Zone — демилитаризованная зона, ДМЗ) — сегмент сети, содержащий общедоступные сервисы и отделяющий их от частных
HDD	(англ. Hard disk drive) твердотельный накопитель без подвижных частей
LDAP	(англ. Lightweight Directory Access Protocol — «легковесный протокол доступа к каталогам») — относительно простой протокол прикладного уровня для доступа к службе каталогов X.500
RAM	(англ. Random Access Memory – память с произвольным доступом) оперативная память, ОЗУ R&D (англ. Research and development — исследования и разработки) этап или отдел в компании, объединяющий несколько подразделений и отвечающий за создание, выведение на рынок продукта и управление его жизненным циклом
SSD	(англ. Solid-State Drive) запоминающее устройство на основе микросхем памяти
VM	Виртуальная машина
ВРМ	Виртуальное рабочее место
Гипервизор	ПО, которое создаёт и запускает виртуальные машины, изолируя операционную систему и ресурсы системы от аппаратного обеспечения
ЦОД	Центр обработки данных или «дата-центр» — это помещение, в котором хранятся компьютеры, серверы и сетевое оборудование
Виртуальная машина (VM)	Виртуальная машина внутри проекта
Центр обработки данных (ЦОД)	Выделенное помещение в котором компания размещает серверное и сетевое оборудование
Внешняя сеть	Конфигурация для подключения ВИ к внешней физической сети

<b>Термин</b>	<b>Описание</b>
HC (health check)	Оценка состояния системы, системных ресурсов
LB (Load Balancer)	Балансировщик нагрузки, предназначенный для распределения сетевого трафика между виртуальными машинами
Snapshot	Моментальный снимок VM
Зона доступности	Один или несколько центров обработки данных, в которых могут быть размещены компоненты облачной инфраструктуры.
Центр сертификации (Certification authority, CA)	Криптографический термин, означающий сторону, чья честность неоспорима, а открытый ключ широко известен. Задача центра сертификации - подтверждать подлинность ключей шифрования с помощью сертификатов электронной подписи.
Чрезмерное выделение ОЗУ (RAM overcommit)	Возможность запускать новые экземпляры виртуальных машин, исходя из действительного использования памяти хоста, а не на основе объема ОЗУ, выделенного для каждого выполняемого экземпляра. Также известно как чрезмерное выделение памяти (memory overcommit).
Шлюз	IP-адрес, как правило, назначаемый маршрутизатору, который передает сетевой трафик другим сетям.
Экстент	в файловых системах непрерывная область носителя информации. Как правило, в файловых системах с поддержкой экстенгов большие файлы состоят из нескольких экстенгов, не связанных друг с другом напрямую.

---

## Дополнительная документация на продукт Гиперус

Дополнительную информации о продукте Гиперус, можно найти в следующих документах.

<b>Обозначение</b>	<b>Наименование</b>
Hyperus_v1.0_AdminGuide	Руководство администратора Гиперус.Инфраструктура

<b>Обозначение</b>	<b>Наименование</b>
Hyperus_v1.0_InstallGuide	Руководство по установке продукта Гиперус.Инфраструктура
Hyperus_v1.0_ProductFeatures	Функциональные возможности продукта Гиперус.Инфраструктура
Hyperus_v1.0_ReleaseNotes	Информация о новой версии продукта Гиперус.Инфраструктура
Hyperus_v1.0_SystemRequirements	Системные требования продукта Гиперус.Инфраструктура
Hyperus_v1.0_LifeCycleMaintenance	Основные сведения о жизненном цикле платформы Гиперус.Инфраструктура

# Требования к составу и квалификации обслуживающего персонала

Данный раздел содержит требования к квалификации и ответственностям специалистов, обеспечивающих эксплуатацию и обслуживание оборудования центра обработки данных (ЦОД) и виртуального ЦОД на платформе ПО Гиперус.Инфраструктура.

- **Системный инженер** – должностное лицо, служебная деятельность которого обеспечивает качественную и безопасную эксплуатацию оборудования ЦОД или виртуального ЦОД – облачной платформы, построенной на основе ПО Гиперус.Инфраструктура – 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, Python, 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 «Гиперус», 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.