BASIS

Руководство по установке Базис.vCore v.1.3

RU.НРФЛ.00007-01.96.01

Москва

04/17/2023

Содержание

Общая информация Базис.vCore v.1.3 4
Аннотация 4
Термины и определения 4
Требования к оборудованию Базис.vCore v.1.3 5
Управляющий узел 5
Вычислительный узел 5
Установка Базис.vCore v.1.3 6
Демонстрационная инсталляция Базис.vCore v.1.3 6
Простая инсталляция Базис.vCore v.1.3 8
Введение 8
Настройка окружения 8
Управляющая часть 8
Вычислительная часть 12
Инсталляция с поддержкой высокой доступности Базис.vCore v.1.3 21
Введение 21
Настройка окружения 21
Управляющая часть 22
Вычислительная часть 32
Интеграция Базис.vCore v.1.3 с Базис.Virtual Security 33
Введение 33
Управляющая часть 33
Вычислительная часть 34
Обновление Базис.vCore v.1.3 35
Простая инсталляция Базис.vCore v1.3 35
Предварительные действия перед обновлением 35
Управляющая часть 35
Вычислительная часть 36
Инсталляция Базис.vCore v.1.3 с поддержкой высокой доступности 37
Предварительные действия перед обновлением 38
Управляющая часть 38
Обновление вычислительной части 40
Обновление vCore 1.2 до 1.3 42
Введение 42
Шаги обновления 42
Удаление Базис.vCore v.1.3 50
Управляющая часть 50
Вычислительная часть 50
Приложения к руководству по установке Базис.vCore v.1.3 51
Приложение 1. Список портов сервисов Базис.vCore v.1.3 51
Приложение 2. Рекомендации по настройке систем хранения для хранилищ Базис.vCore v.1.3 52

Введение 52
NFS 52
iSCSI 53
Fibre Channel 54

Общая информация Базис.vCore v.1.3

Аннотация

Настоящая инструкция предназначена для технического администратора ПО и содержит подробное описание работ, необходимых для установки ПО.

Термины и определения

Термин, сокращение	**Сокращение**	**Определение**
Базис vCore	vCore	Программный комплекс управления масштабируемой системой виртуализации с высокой надежностью.
Вычислительный узел	ВУ	Структурная единица, представляющая физический вычислительный узел, на котором размещен гипервизор и агент vCore.
Гипервизор		Компонент, предоставляющий функции виртуализации, мониторинга, управления сетью и дисками виртуальных машин.
Дата-центр	ДЦ	Структурная единица, представляющая абстракцию физического дата-центра. Объединяет под собой хранилища, кластеры, сети, шаблоны виртуальных машин. Дата-центр может проецироваться как на физический дата-центр, так и просто на группу нескольких серверов (серверную стойку), которые имеют общий доступ к определенным хранилищам и сетям.
Домен		Структурная единица верхнего уровня, которая объединяет под собой ведение объектов безопасности (пользователи, роли, группы, разрешения), инфраструктурных объектов (дата-центры, хранилища, сети, кластеры и т.д.), и объектов учета ресурсов облака (предложений развертки, проектов, виртуальных машин).
Инфраструктура		Совокупность компьютерного оборудования (серверы, системы хранения данных, коммутационное оборудование и др.) и системного ПО, включая средства виртуализации.
Кластер		Структурная единица, которая объединяет под собой вычислительные узлы одной архитектуры (тип и семейство процессора).
Программное обеспечение	ПО	Совокупность машинного кода, выполняемых вычислительной системой.
Ресурс		Комплекс программно-аппаратных средств облачной платформы.
Управляющий узел/ контроллер		Узел, осуществляющий управление системой виртуализацией. На узле располагаются коммуникационные сервисы, которые связывают компоненты системы виртуализации и исполняет функции сервиса очередей сообщений, базы данных и панели управления, а так же может содержать компоненты балансировки нагрузки.

Требования к оборудованию Базис.vCore v.1.3

Управляющий узел

Количество управляющих узлов - 1.

Операционная система - Fedora 35.

Требования по аппаратным ресурсам:

Параметр	Минимальные	Рекомендованные
Процессор	8 ядер ЦП 2.4 Ггц	16 ядерЦП 2.4 Ггц
ОЗУ	16 ГБ	24 ГБ и более
Диск	2 x 100 ГБ в режиме RAID1	2 x 100 ГБ и более в режиме RAID1
Сеть	Пропускная способность не менее 1 Гбит/с.	Пропускная способность не менее 1 Гбит/с.

Требования по архитектуре процессора: x86_64. Поддерживаемые модели:

- AMD: с микроархитектурой Zen и выше.
- Intel: с микроархитектурой Skylake (6-го поколения) и выше.

Вычислительный узел

Минимально необходимое количество вычислительных узлов - 1.

Требования по аппаратным ресурсам:

Параметр	Минимальные	Рекомендованные
Процессор	8 ядер ЦП 2.4 Ггц	16 ядер ЦП 2.4 Ггц
ОЗУ	32 ГБ	64 ГБ и более
Диск	100 ГБ	100 ГБ и более
Сеть	Пропускная способность не менее 1 Гбит/с.	Пропускная способность не менее 1 Гбит/с.

Диск в ВУ необходим, если требуется полноценная установка гипервизора на вычислительный узел и для локального сохранения конфигурации ВУ.

На данный момент поддерживаются процессоры двух типов: AMD64 и Intel64. Процессоры должны иметь поддержку аппаратной виртуализации AMD-V или Intel VT. Также необходимо наличие атрибута NX bit[1].

Поддерживаемые модели процессоров:

- AMD: Семейство процессоров EPYC с микроархитектурой Zen и выше.
- Intel: Семейство процессоров Xeon с микроархитектурой Skylake (6-го поколения) и выше.

1 https://ru.wikipedia.org/wiki/NX_bit

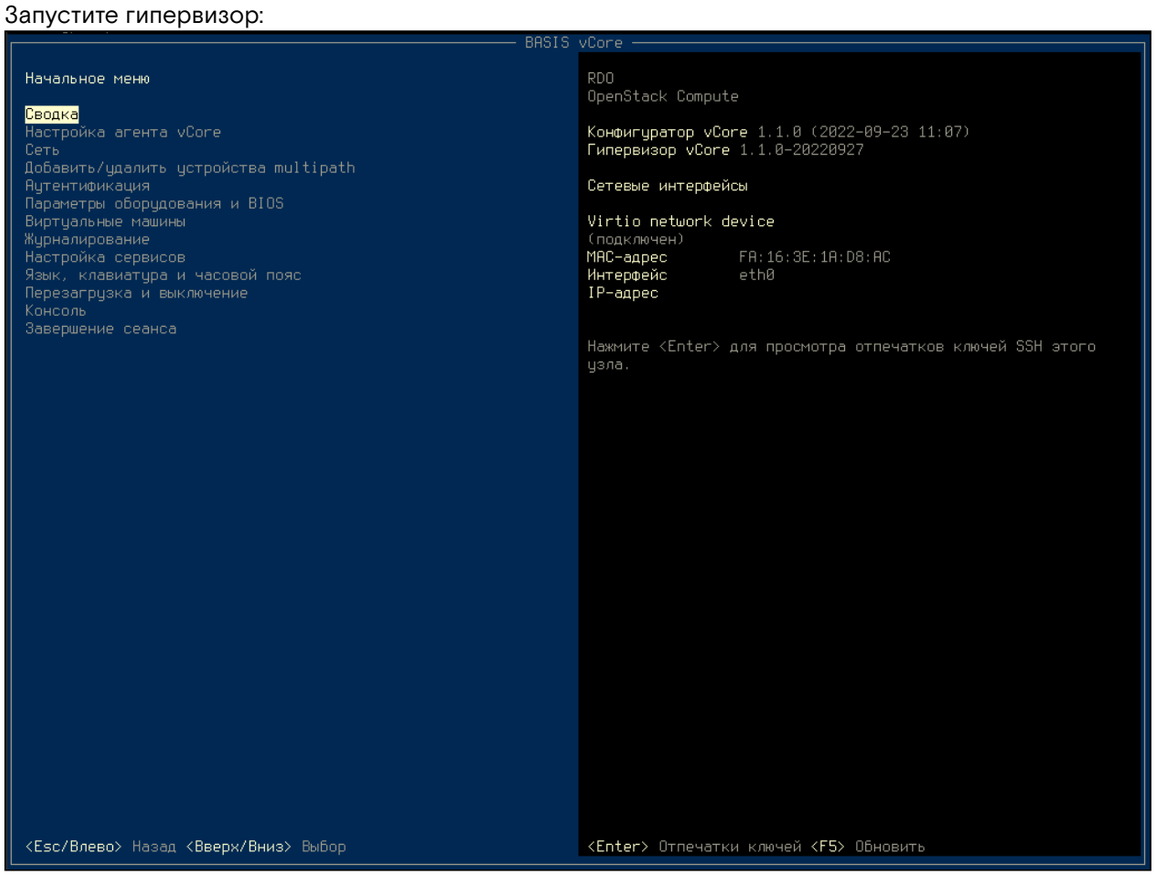
Установка Базис.vCore v.1.3

Демонстрационная инсталляция Базис.vCore v.1.3

Демонстрационная инсталляция платформы осуществляется на одном физическом узле. Дистрибутив предоставляется в ISO-образа, который включает в себя компоненты управляющей и вычислительной части.

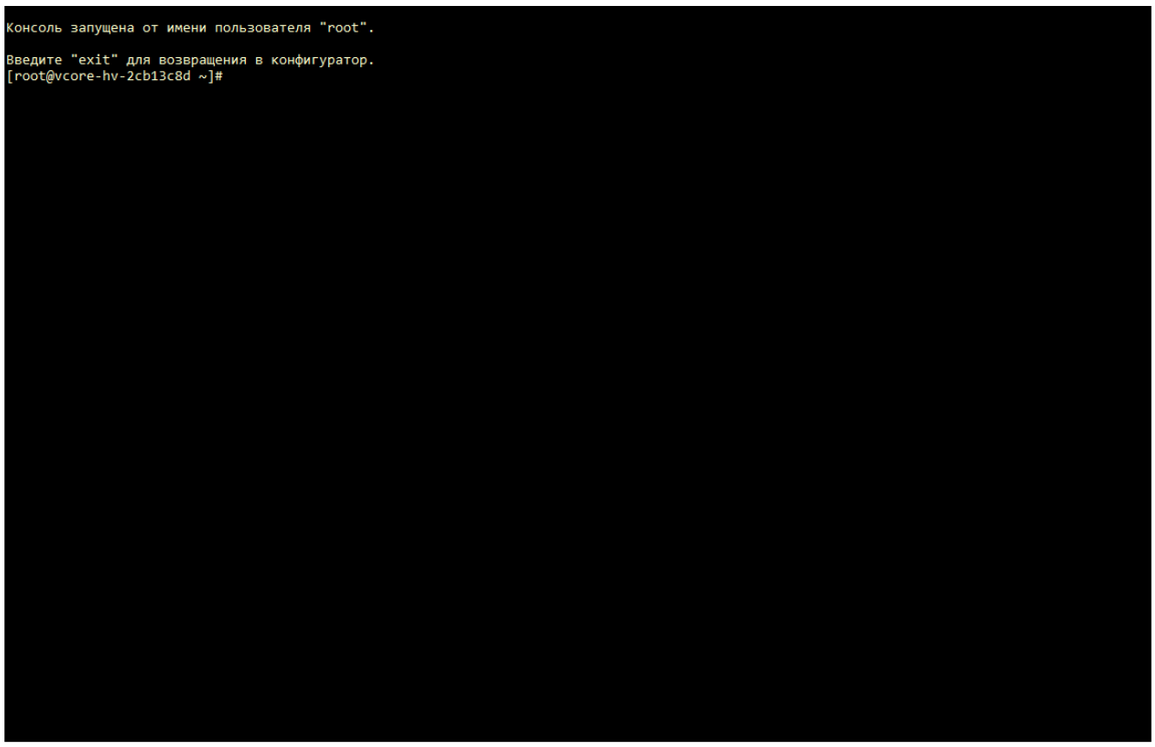
> **Важно**
>
> Образ демонстрационной инсталляции предоставляется отдельно по запросу.

1. Скачайте и установите ISO-образ.
2. Запустите гипервизор:



Начальное меню

3. Перейдите в раздел "Консоль":



Консоль гипервизора

4. Откройте файл конфигурации Broker - `/opt/vcore/broker/config/application.properties`:

```
nano /opt/vcore/broker/config/application.properties
```

5. Укажите параметры:

```
quarkus.http.host=BROKER_IP_DNS_ADDR
provider.storage.bind-address=BROKER_IP_DNS_ADDR
provider.storage.static-discovery-addresses=BROKER_IP_DNS_ADDR:47500..47510
provider.storage.jdbc-address=BROKER_IP_DNS_ADDR
provider.transport.bind-address=BROKER_IP_DNS_ADDR
provider.transport.static-discovery-addresses=tcp://BROKER_IP_DNS_ADDR:61617
provider.transport.use-multicast-discovery=false
```

Где: BROKER_IP_DNS_ADDR - IP-адрес или DNS-имя узла, на котором установлен образ.

> **Важно**
>
> При запуске образа в сетях с Dual Stack необходимо использовать IPv4-адреса в конфигурации.

6. Запустите Broker при помощи команды:

```
systemctl start vcore-broker
```

7. Откройте файл конфигурации Control - `/opt/vcore/control/config/application.properties`:

```
nano /opt/vcore/control/config/application.properties
```

8. Укажите параметры:

```
quarkus.http.host=CONTROL_IP_DNS_ADDR
quarkus.datasource.jdbc.url=jdbc:ignite:thin://CONTROL_IP_DNS_ADDR?
distributedJoins=true
broker-rest-api/mp-rest/url=http://CONTROL_IP_DNS_ADDR:8081/api
quarkus.artemis.url=tcp://CONTROL_IP_DNS_ADDR:61616?
clientFailureCheckPeriod=5000&retryInterval=1000&reconnectAttempts=5
```

Где: CONTROL_IP_DNS_ADDR - IP-адрес или DNS-имя узла, на котором установлен образ.

9. Запустите Control при помощи команды:

```
systemctl start vcore-control
```

10. Откройте файл конфигурации Agent - `/opt/vcore/agent/config/application.properties:`

```
nano /opt/vcore/agent/config/application.properties
```

11. Укажите параметры:

```
quarkus.http.host=AGENT_IP_DNS_ADDR
quarkus.artemis.url=tcp://AGENT_IP_DNS_ADDR:61616?
clientFailureCheckPeriod=5000&retryInterval=1000&reconnectAttempts=5&callTimeout=3
000
```

Где: AGENT_IP_DNS_ADDR - IP-адрес или DNS-имя узла, на котором установлен образ.

12. Запустите Agent при помощи команды:

```
systemctl start vcore-agent
```

13. Подключитесь к панели управления. В адресной строке браузера введите DNS-имя или IP-адрес узла в формате - https://IP_DNS_ADDR:8082[2].

Простая инсталляция Базис.vCore v.1.3

- Введение (см. стр. 8)
- Настройка окружения (см. стр. 8)
- Управляющая часть (см. стр. 8)
- Вычислительная часть (см. стр. 12)

Введение

Установка ПО осуществляется при помощи пакетов RPM.

Процесс установки включает:

1. Установка компонентов системы управления.
2. Установка и настройка вычислительного узла с агентом гипервизора.

Настройка окружения

Пакеты vCore расположены в репозитории проекта. Для включения этого репозитория нужно сделать следующее.

1. Создайте файл `/etc/yum.repos.d/vcore-engine.repo`.
2. В файле `/etc/yum.repos.d/vcore-engine.repo` укажите следующее:

```
[vcore-engine]
name=vCore Engine
baseurl=https://vcore-public:vcore-public@maven.tionix.ru/artifactory/vcore-rpm/
fedora/linux/$releasever/$basearch
enabled=1
gpgcheck=0
```

3. Запустите команду:

```
dnf update
```

Управляющая часть

Установка компонентов системы управления

Система управления в vCore состоит из двух компонентов: Broker и Control.

2 https://IP_DNS_ADDR:8082:

- Broker - является основным компонентом проекта и обеспечивает выполнение двух основных функций: функцию хранения состояния платформы в базе данных Apache Ignite и функцию сервиса сообщений.
- Control - предоставляет инструмент управления платформой в виде веб-приложения и иных функций. Control зависим от состояния сервиса Broker.

Установка Broker:

```
dnf install vcore-broker
```

Установка Control:

```
dnf install vcore-control
```

Приложения Control и Broker будут установлены в каталогах: `/opt/vcore/control` и `/opt/vcore/broker` соответственно.

Настройка сервисов

По умолчанию сервисы слушают только loopback-интерфейс. При отсутствии прокси-сервера или балансировщика нагрузки можно поменять этот адрес, чтобы получить доступ до, например, веб-интерфейса.

Broker

Особенности настройки:

- Путь до конфигурации Broker: `/opt/vcore/broker/config/application.properties`
- Стандартные порты (основные): 8081/tcp, 61616/tcp (messaging), 10800/tcp (jdbc)

Основные настройки

Укажите IP-адрес или DNS-имя узла брокера в следующие параметры:

```
quarkus.http.host=BROKER_IP_DNS_ADDR
provider.storage.bind-address=BROKER_IP_DNS_ADDR
provider.storage.jdbc-address=BROKER_IP_DNS_ADDR
provider.transport.bind-address=BROKER_IP_DNS_ADDR
```

Для обнаружения сервисов брокер может использовать два сетевых механизма: *multicast* для динамической конфигурации и *unicast* для статической. По умолчанию рекомендуется использовать multicast.

Настройка multicast

Укажите следующие параметры в конфигурацию брокера:

```
provider.storage.use-multicast-discovery=true
provider.storage.multicast-interface-addresses=BROKER_IP_DNS_ADDR
provider.transport.use-multicast-discovery=true
provider.transport.multicast-interface-address=BROKER_IP_DNS_ADDR
```

Настройка unicast

Укажите следующие параметры в конфигурацию брокера:

```
provider.storage.use-multicast-discovery=false
provider.storage.static-discovery-addresses=BROKER_IP_DNS_ADDR:47500..47510
provider.transport.use-multicast-discovery=false
provider.transport.static-discovery-addresses=tcp://BROKER_IP_DNS_ADDR:61617
```

Укажите логин/пароль для взаимодействия сервисов. Значения используемые по умолчанию:

```
quarkus.artemis.username=user
quarkus.artemis.password=Q0hBTkdFTUUhIQ==
```

Задайте необходимые значения:

```
quarkus.artemis.username=login
quarkus.artemis.password=password
```

Где:

- login - Логин.
- password - Пароль в формате base64.

Control

Особенности настройки:

- Путь до конфигурации Control: `/opt/vcore/control/config/application.properties`
- Стандартный порт: 8082/tcp.

Для Control нужно поменять адрес прослушивания сервиса quarkus:

```
quarkus.http.host=CONTROL_IP_DNS_ADDR
```

Укажите адрес подключения к СУБД Ignite:

```
quarkus.datasource.jdbc.url=jdbc:ignite:thin://BROKER_IP_DNS_ADDR?distributedJoins=true
```

Укажите адрес Broker API:

```
broker-rest-api/mp-rest/url=http://BROKER_IP_DNS_ADDR:8081/api
```

Укажите адрес службы очереди сообщений:

- при варианте multicast нужно указать адрес, прописанный в параметре *provider.transport.multicast-group* конфигурации брокера:

  ```
  quarkus.artemis.url=udp://231.7.7.7:9876?
  clientFailureCheckPeriod=5000&retryInterval=1000&reconnectAttempts=5
  ```

- при варианте unicast нужно указать адрес самого брокера:

  ```
  quarkus.artemis.url=tcp://BROKER_IP_DNS_ADDR:61616?
  clientFailureCheckPeriod=5000&retryInterval=1000&reconnectAttempts=5
  ```

Укажите логин/пароль для взаимодействия сервисов. Значения используемые по умолчанию:

```
quarkus.artemis.username=user
quarkus.artemis.password=Q0hBTkdFTUUhIQ==
```

Задайте необходимые значения:

```
quarkus.artemis.username=login
quarkus.artemis.password=password
```

Где:

- login - Логин.
- password - Пароль в формате base64.

Запуск сервисов

> **Важно**
>
> При запуске сервисов в сетях с Dual Stack необходимо использовать IPv4-адреса в конфигурации.

Для запуска компонентов системы управления достаточно старта нужных юнитов systemd:

```
systemctl enable vcore-control vcore-broker
systemctl start vcore-broker vcore-control
```

Control и Broker могут достаточно долго инициализироваться, поэтому в systemd включен режим нотификации готовности сервиса к работе.

Примечание

При использовании FirewallD в качестве системы управления Firewall необходимо открыть порты сервисов. Для этого используйте следующие команды:

```
firewall-cmd --zone=public --add-service=vcore-broker --permanent
firewall-cmd --zone=public --add-service=vcore-control --permanent
firewall-cmd --zone=public --add-service=vcore-balancer --permanent
```

Перезапустите Firewall:

```
firewall-cmd --reload
```

Для проверки портов выполните:

```
nc -v IP_ADDR PORT
```

Где:

- IP_ADDR - IP-адрес узла;
- PORT - Порт сервиса.

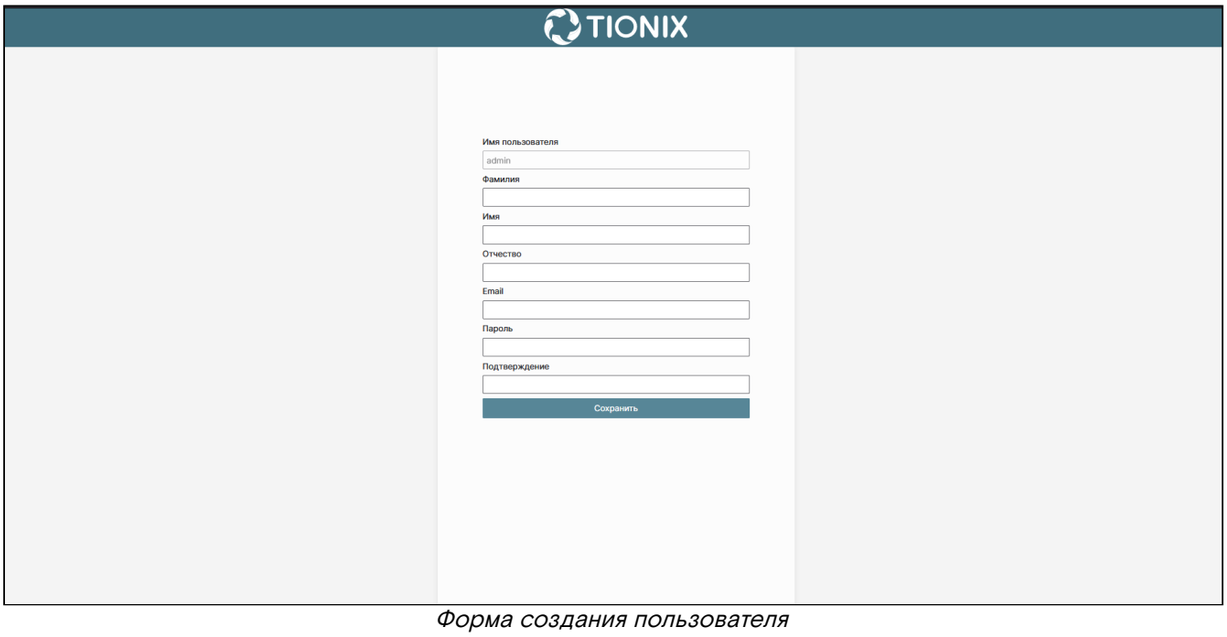
Список портов сервисов по умолчанию доступен по ссылке (см. стр. 51).

Настройка панели управления

После запуска Control по адресу http://CONTROL_IP_DNS_ADDR:8082 будет доступен веб-интерфейс платформы. Укажите первичные данные для пользователя admin:

Информация

Для корректной работы системы рекомендуется использовать Google Chrome или Firefox актуальных версий.



Форма создания пользователя

Заполните поля формы создания пользователя. Подтвердите создание кнопкой "Сохранить". После этого укажите следующие данные для входа в систему:

- домен: master;
- пользователь: admin;
- пароль: указанный для пользователя admin.

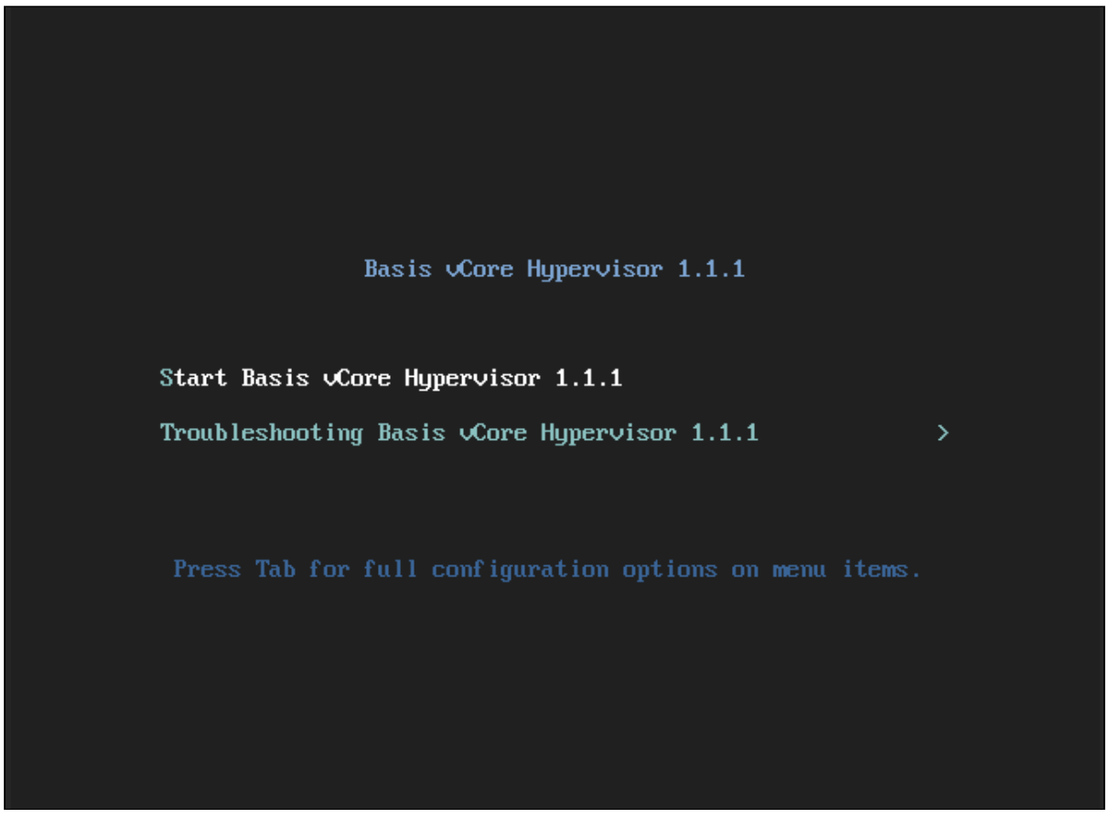
Вычислительная часть

Установка гипервизора

Образ гипервизора в формате ISO доступен по этой ссылке[3].

Загрузите образ гипервизора в систему. Образ гипервизора предоставляется отдельно.

Запустите систему из полученного образа:



Меню запуска системы

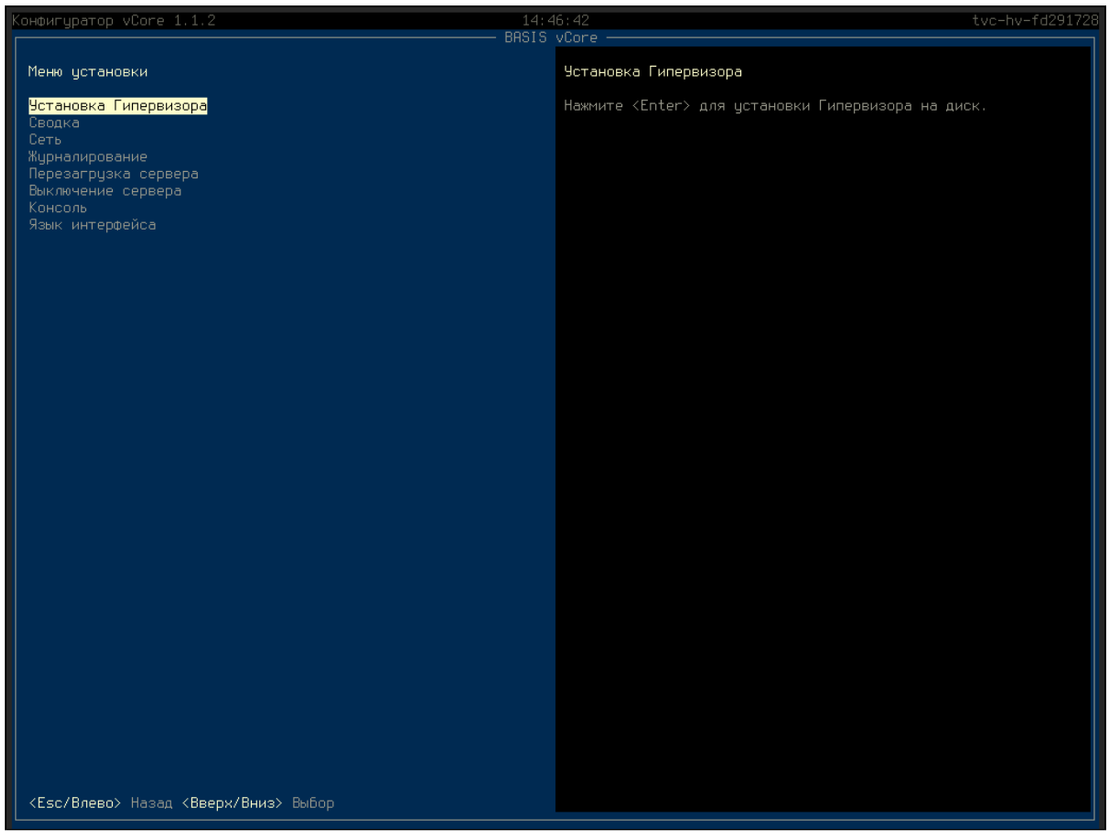
Выберите первый пункт запуска системы.

Дождитесь завершения загрузки системы и ознакомьтесь с лицензионным соглашением:

3 https://vcore-public:vcore-public@maven.tionix.ru:443/artifactory/vcore-hyper-os/

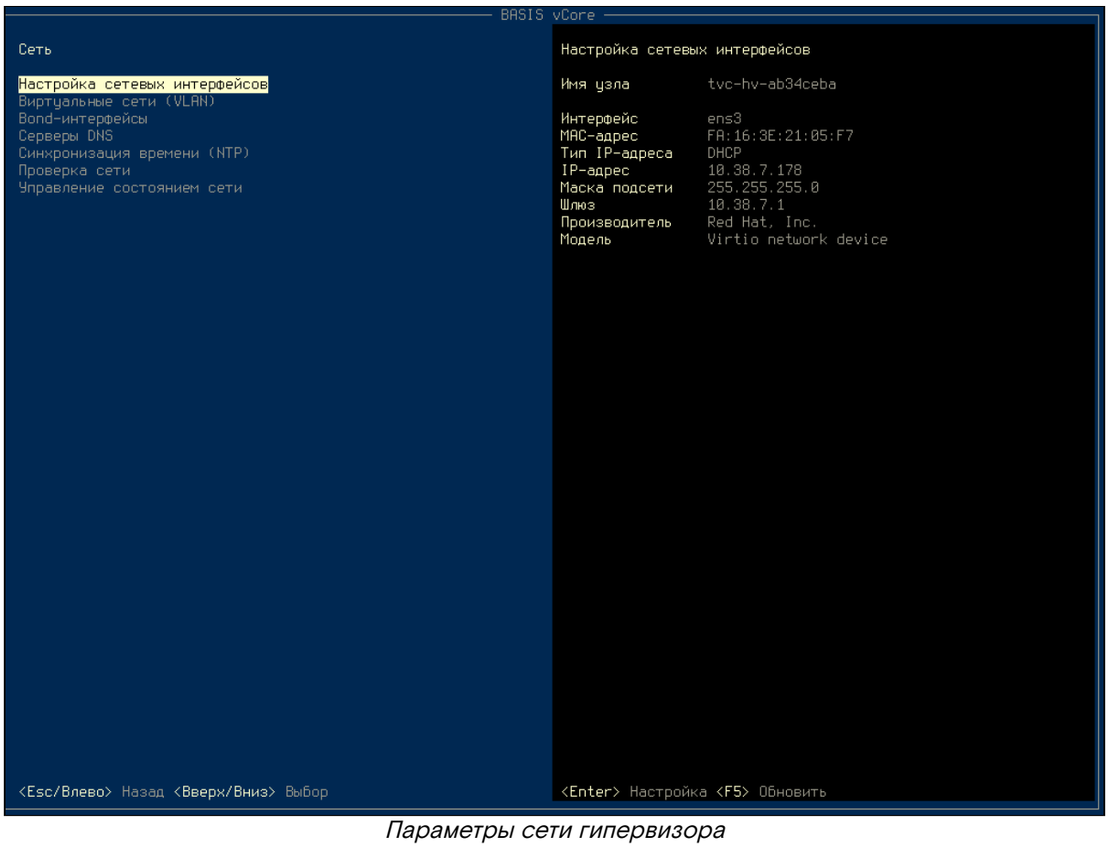
Лицензионное соглашение Базис vCore

Примите лицензионное соглашение при помощи клавиши Enter и перейдите в главное меню:



Главное меню гипервизора

В открывшейся вкладке с отобразившимся меню установки, в первую очередь ознакомьтесь с конфигурацией сети по умолчанию. Для этого выберите опцию меню установки "Сеть" и перейдите в раздел «Сеть», нажав клавишу Enter:



Параметры сети гипервизора

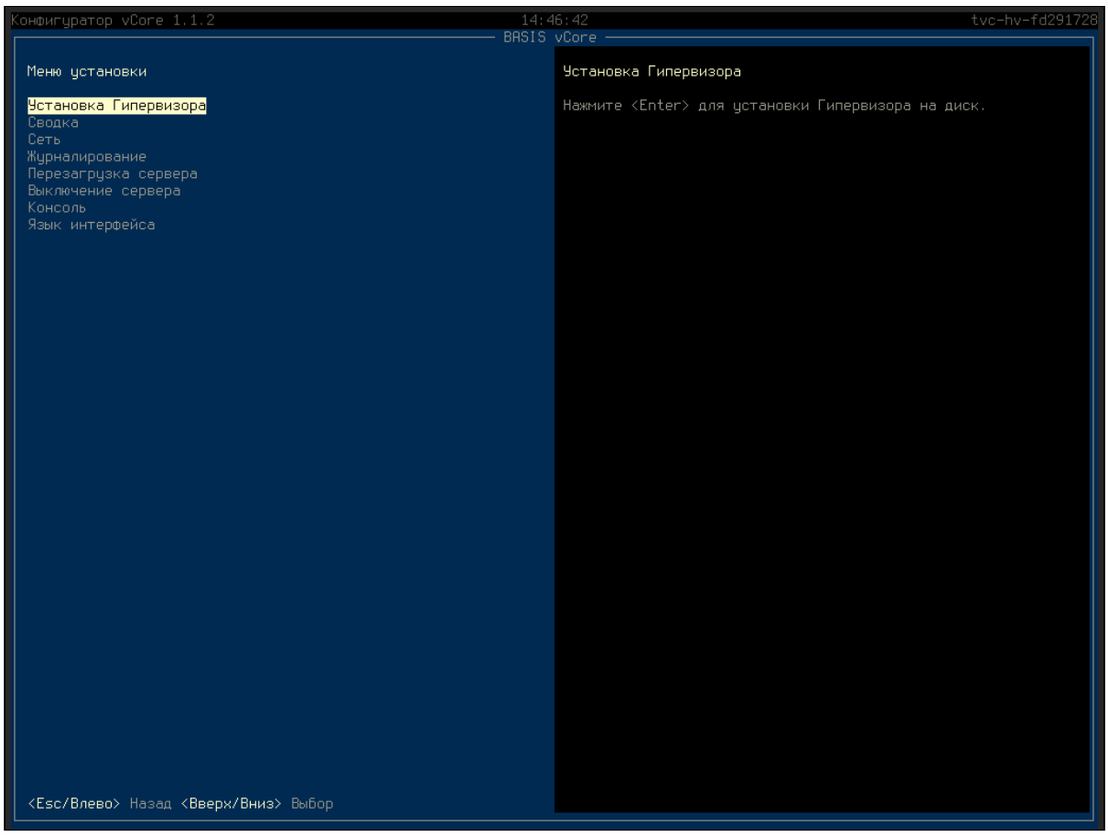
При необходимости установите следующие параметры:

- Настройка сетевых интерфейсов;
- Виртуальные сети (VLAN);
- Bond-интерфейсы;
- Серверы DNS;
- Синхронизация времени (NTP);
- Проверка сети;
- Управление состоянием сети.

Важно

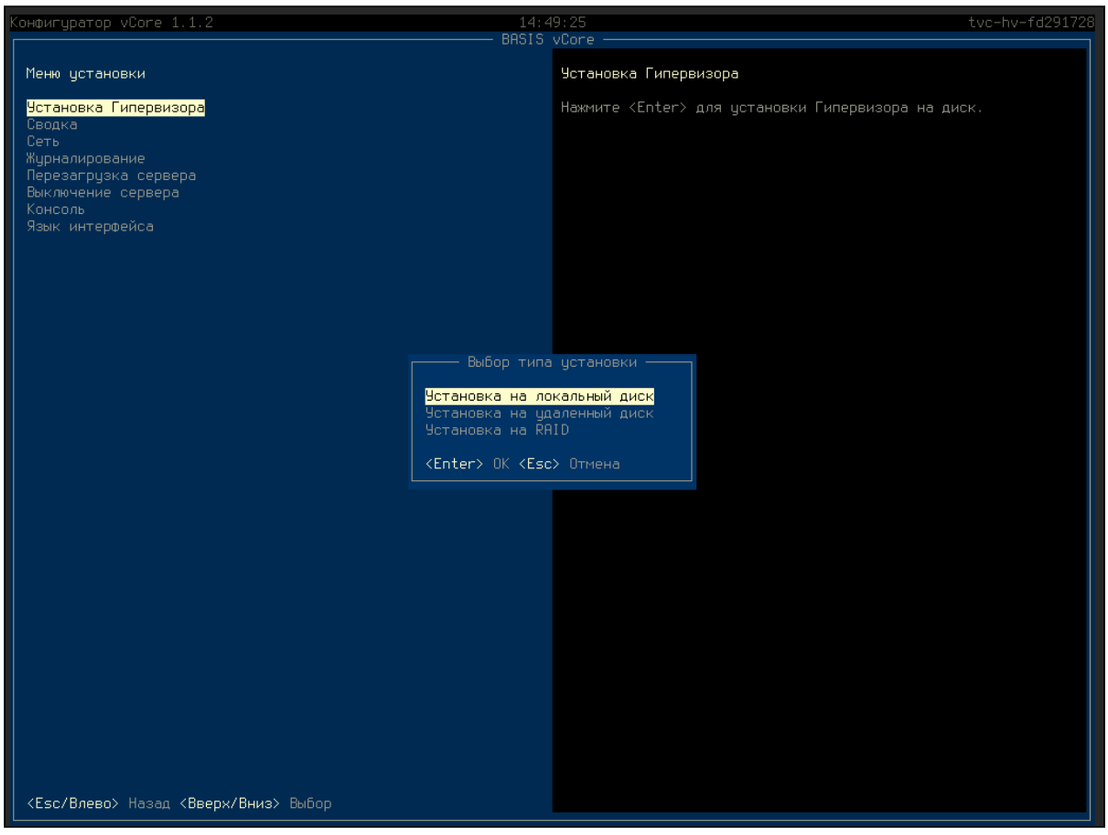
При запуске гипервизора в сети с Dual Stack необходимо использовать IPv4-адреса в конфигурации.

Для установки гипервизора выберите пункт «Установка Гипервизора»:



Меню установки гипервизора

Выберите тип установки:

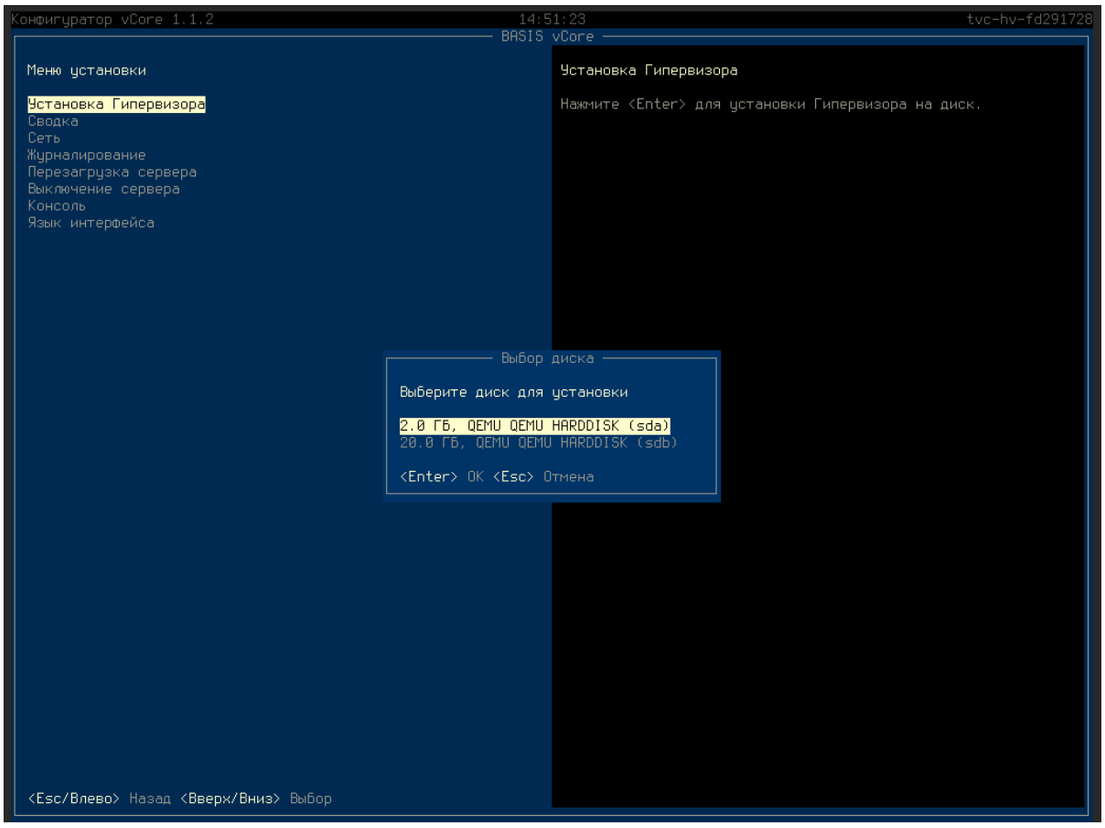


Меню выбора типа установки

Поддерживаемые типы:

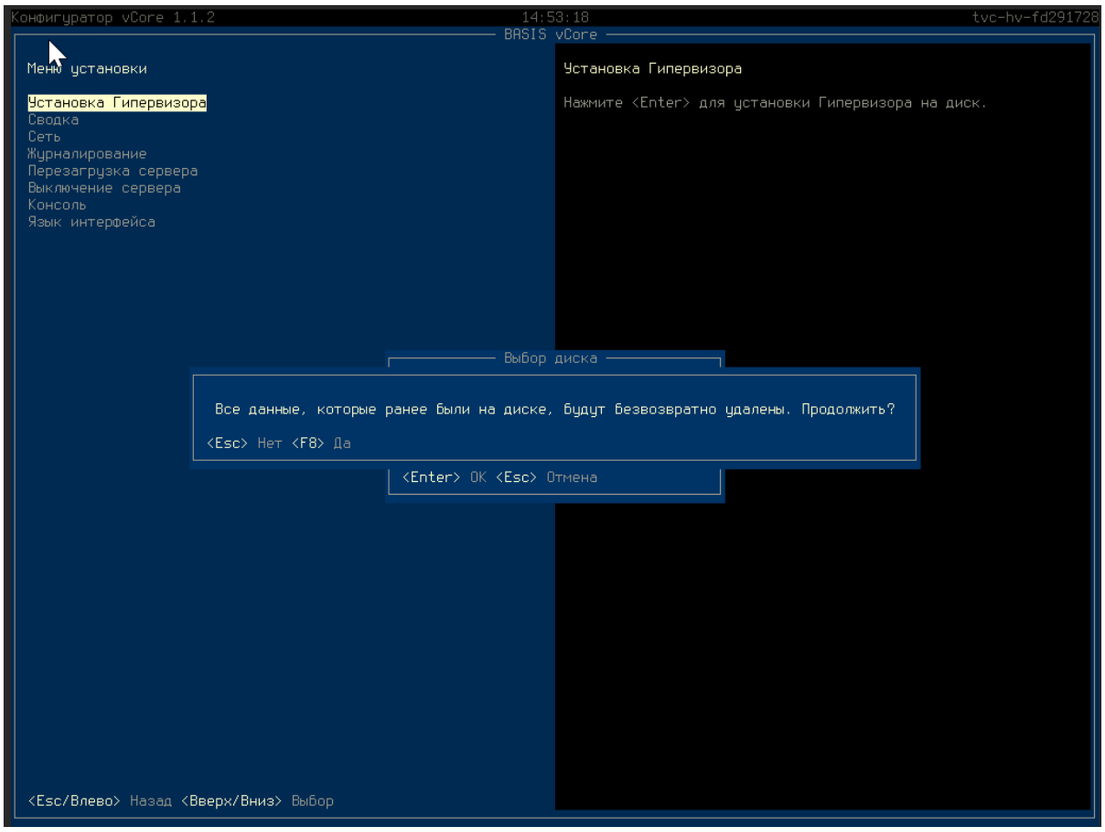
- Установка на локальный диск;
- Установка на удаленный диск;
- Установка на RAID. Поддерживаемый тип – RAID 1.

Выберите диск для установки:



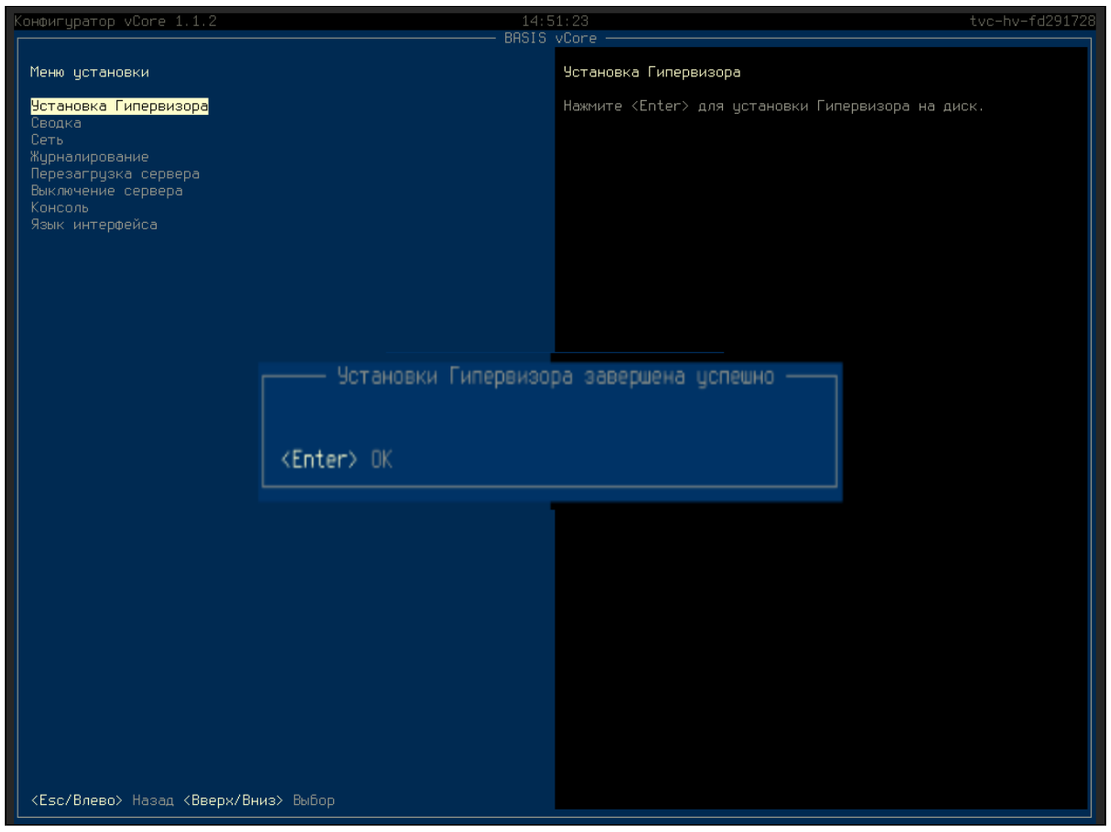
Меню выбора диска

Подтвердите очистку выбранного диска:



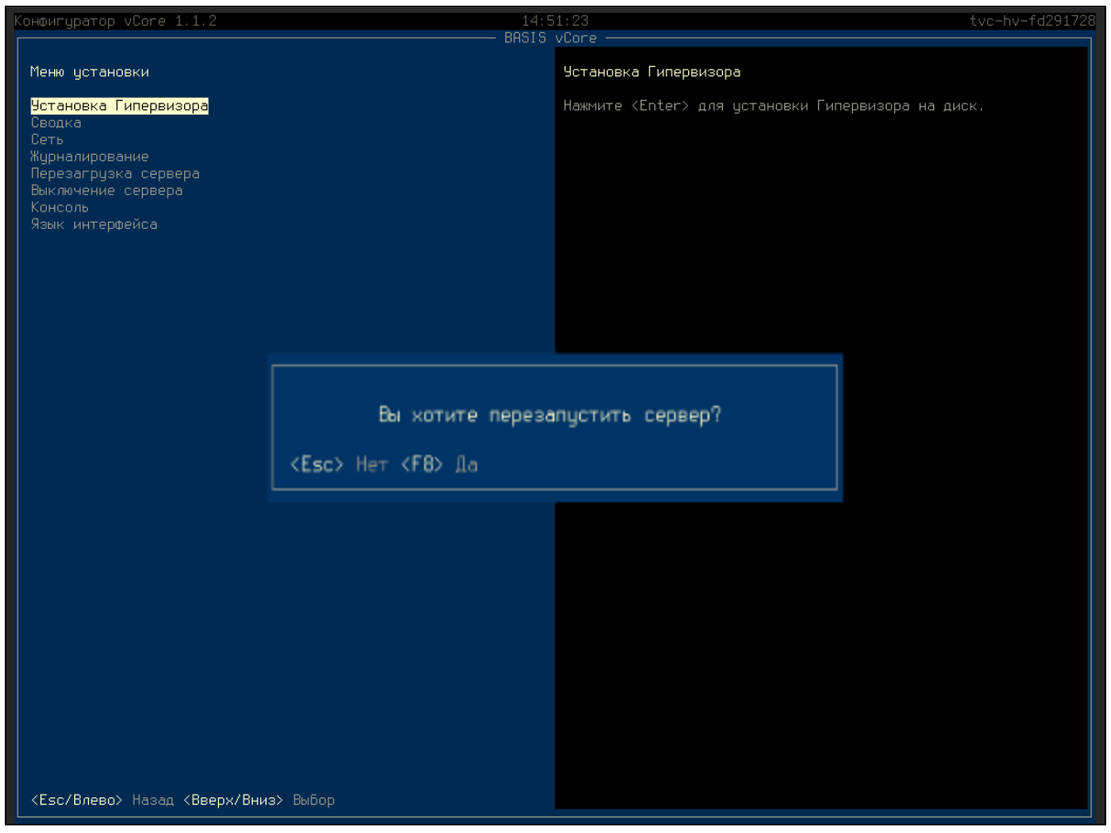
Подтверждение очистки диска

Дождитесь завершения процесса установки:



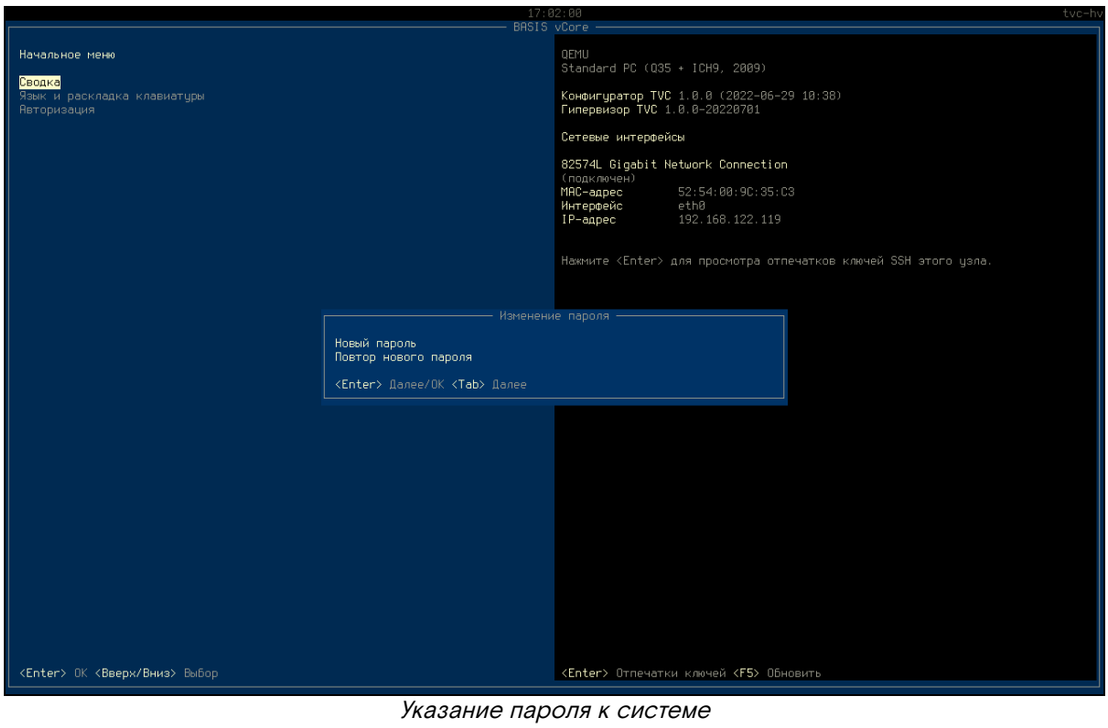
Сообщение о завершении процесса установки

Подтвердите перезагрузку системы:



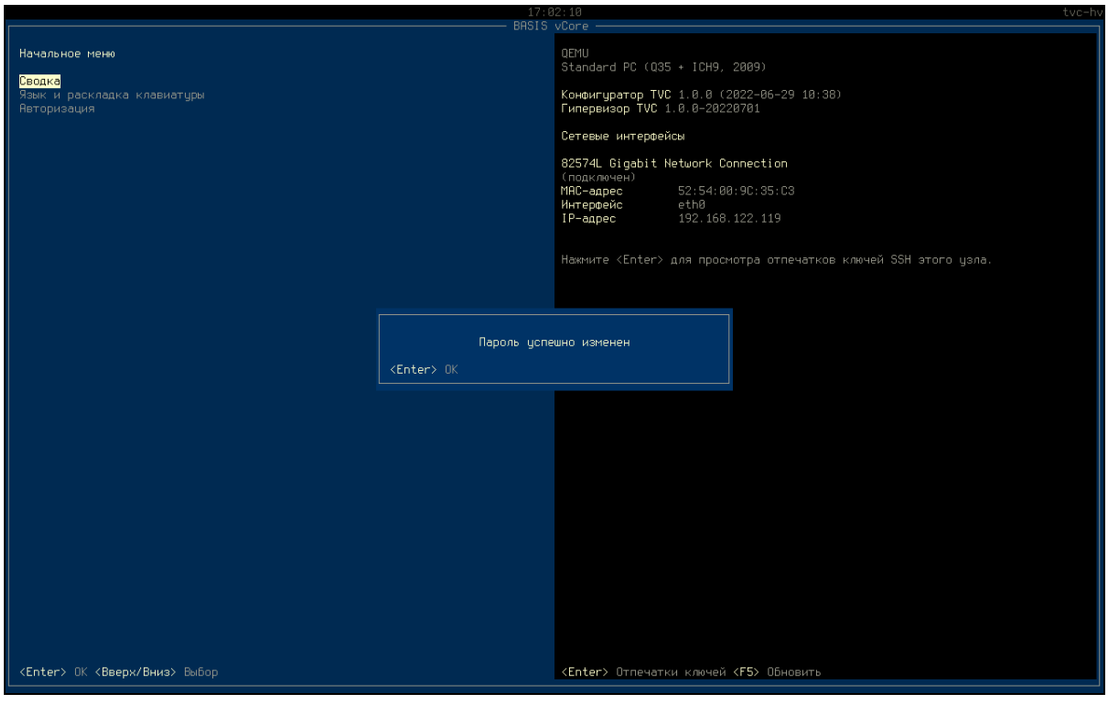
Подтверждение перезагрузки системы

После завершения процесса перезагрузки необходимо установить пароль к системе:



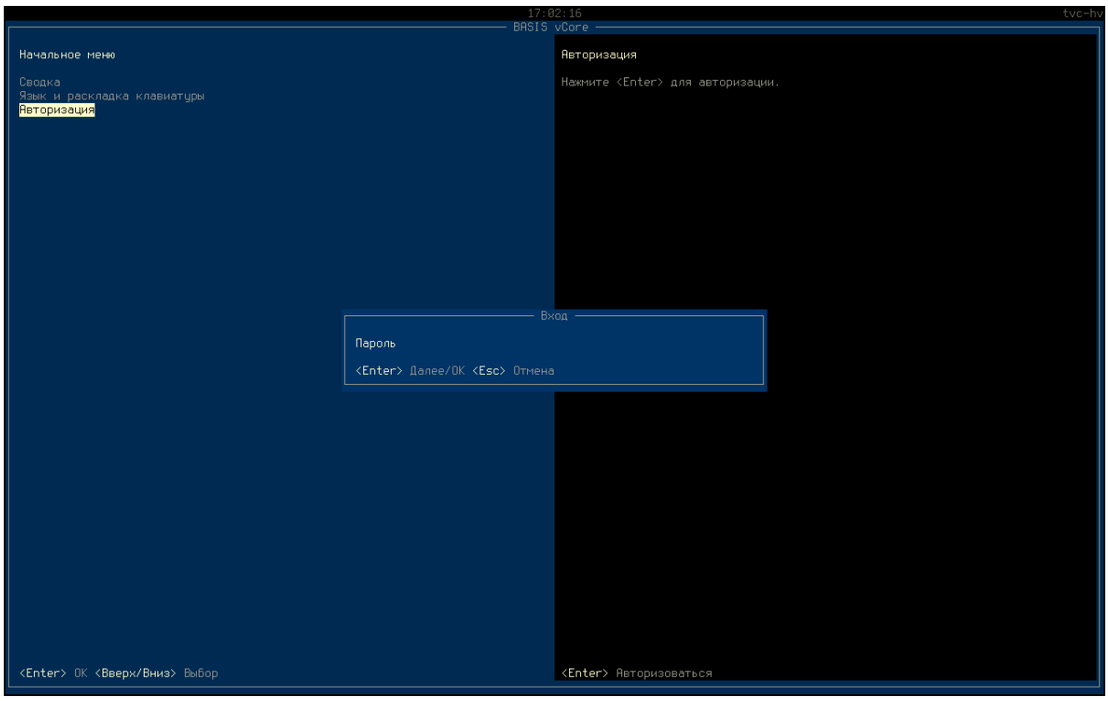
Указание пароля к системе

Убедитесь, что пароль успешно изменен:



Сообщение об успешном изменении пароля

Перейдите в меню "Авторизация" и укажите пароль для входа в систему:



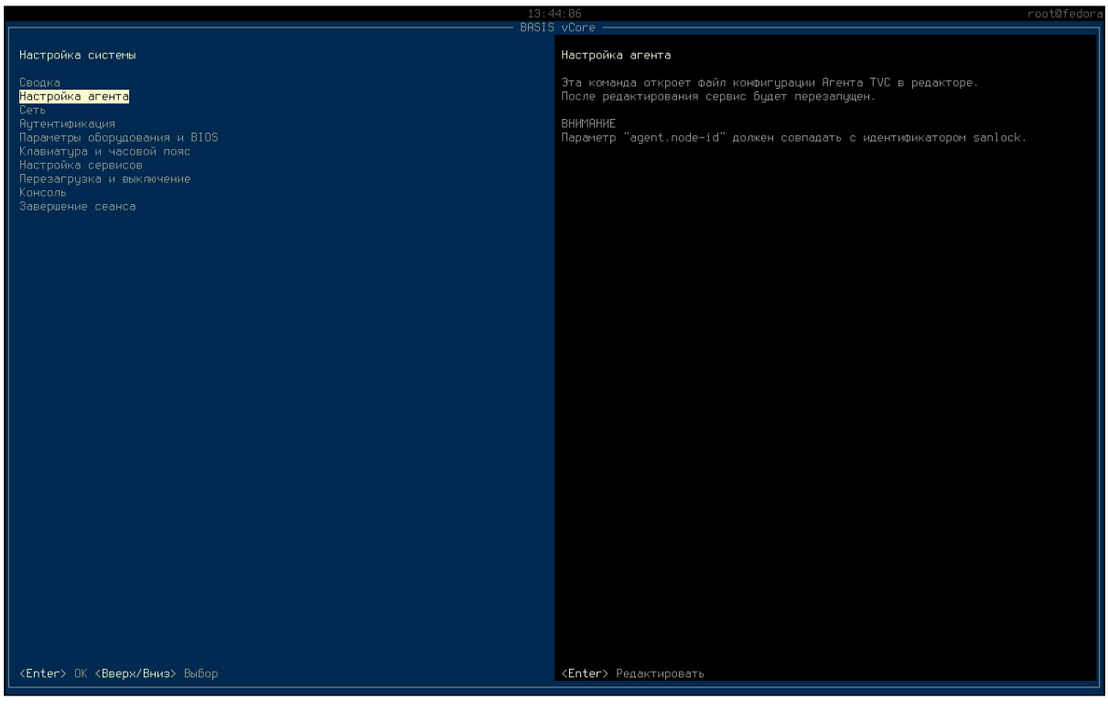
Окно входа в систему

Установка гипервизора завершена, доступны все меню настроек системы.

Настройка агента гипервизора

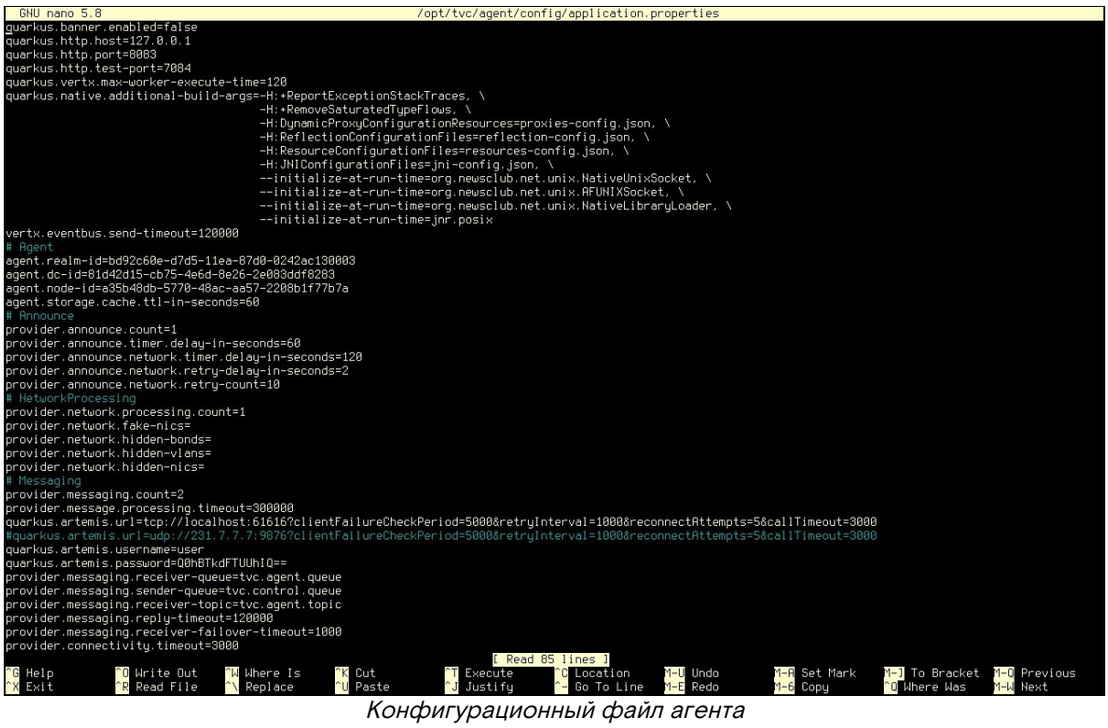
> ⚠ Перед добавлением гипервизора в vCore убедитесь в том, что в систему добавлен пользователь admin. В противном случае агент не сможет найти данные основного домена master и закончит работу с ошибкой подключения к AMQP.

После перезагрузки системы откроется меню управления гипервизором. Перейдите в меню «Настройка агента»:



Меню настройки гипервизора

Нажмите клавишу Enter:



Конфигурационный файл агента

В открывшемся файле конфигурации укажите следующие параметры.

Вначале настройте адрес самого агента:

```
quarkus.http.host=AGENT_IP_DNS_ADDR
```

Затем укажите адрес службы очереди сообщений:

- при варианте multicast нужно указать адрес прописанный в параметре *provider.transport.multicast-group* конфигурации брокера, для подключения по протоколу UDP:

  ```
  quarkus.artemis.url=udp://231.7.7.7:9876?
  clientFailureCheckPeriod=5000&retryInterval=1000&reconnectAttempts=5
  ```

- при варианте unicast нужно указать адрес самого брокера для подключения по протоколу TCP:

  ```
  quarkus.artemis.url=tcp://BROKER_IP_DNS_ADDR:61616?
  clientFailureCheckPeriod=5000&retryInterval=1000&reconnectAttempts=5
  ```

После сохранения агент будет автоматически перезагружен.

Настройка идентификаторов агента и сервиса sanlock

Для корректной работы сетевых блочных устройств в вычислительных узлах необходимо, что сервис sanlock имел ID, который равен идентификатору агента в параметре agent.node-id. Для каждого вычислительного узла этот идентификатор должен быть уникальным.

Сгенерируйте уникальный UUID на своей локальной машине или на вычислительном узле через функцию доступа к консоли:

```
uuidgen
```

Сохраните полученную комбинацию UUID. Для примера: c422b2fd-5bc9-4d53-ad65-d06c595a5f67

В конфигураторе выберите пункт "Настройка агента" и в ней укажите сгенерированную комбинацию UUID в параметр agent.node-id:

```
agent.node-id=c422b2fd-5bc9-4d53-ad65-d06c595a5f67
```

Войдите в Консоль, откройте файл /etc/sysconfig/sanlock и в ней укажите комбинацию для параметра AGENT_ID:

```
AGENT_ID=c422b2fd-5bc9-4d53-ad65-d06c595a5f67
```

Перезапустите сервис sanlock:

```
systemctl restart sanlock
```

На этом настройка вычислительной части завершена.

> **Важно**
>
> После установки и настройки на вычислительном узле ПО "Базис.vCore" узел автоматически подключится к платформе со статусом "Обслуживание". Для ввода узла в эксплуатацию с таким состоянием его необходимо активировать, добавив его в один из кластеров. Данная процедура подробна описана в руководстве Администратора vCore.

Инсталляция с поддержкой высокой доступности Базис.vCore v.1.3

- Введение (см. стр. 21)
- Настройка окружения (см. стр. 21)
  - Fedora (см. стр. 21)
  - Almalinux (см. стр. 22)
- Управляющая часть (см. стр. 22)
  - Установка системы управления кластером Pacemaker (см. стр. 22)
  - Настройка виртуального IP-адреса (см. стр. 23)
  - Установка балансировщика нагрузки (см. стр. 24)
    - Установка сервиса balancer (см. стр. 24)
    - Настройка ресурса balancer (см. стр. 28)
  - Установка и настройка брокера (см. стр. 29)
  - Установка и настройка контрола (см. стр. 31)
  - Настройка панели управления (см. стр. 32)
- Вычислительная часть (см. стр. 32)

Введение

Установка Базис vCore в режиме высокой доступности подразумевает наличие нескольких управляющих (минимум 3) и вычислительных узлов (минимум 3). Процесс установки включает в себя:

- Установка компонентов системы управления:
  - **Broker** - является "сердцем" проекта и предоставляет две основные функции: функции хранения состояния платформы в базе данных Apache Ignite и функции брокера сообщений.
  - **Control** - предоставляет инструмент управления платформой в виде веб-приложения и иных функций. Control зависим от состояния сервиса Broker.
  - **Balancer** - обладает средствами балансировки нагрузки для сервисов control и broker.
- Установка и настройка гипервизора.

Все шаги по установке и настройке необходимо выполнять на всех узлах кластера, если не указано иное.

Настройка окружения

Пакеты vCore расположены в репозитории Basis.vCore. Для включения этого репозитория нужно сделать следующее.

Fedora

1. Создайте файл `/etc/yum.repos.d/vcore-hyper.repo`.
2. В файле `/etc/yum.repos.d/vcore-hyper.repo` укажите следующее:

```
[vcore-engine]
name= vCore Engine $releasever
baseurl=https://vcore-public:vcore-public@maven.tionix.ru/artifactory/vcore-rpm/
fedora/linux/$releasever/$basearch
enabled=1
```

```
gpgcheck=0
```

3. Для обновления системы запустите команду:

```
dnf update
```

Almalinux

1. Создайте файл `/etc/yum.repos.d/vcore-hyper.repo`.
2. В файле `/etc/yum.repos.d/vcore-hyper.repo` укажите следующее:

```
[vcore-engine]
name=  vCore Hyper Engine AL8
baseurl=https://vcore-public:vcore-public@maven.tionix.ru/artifactory/vcore-rpm/
almalinux/8/vcore/x86_64/
enabled=1
gpgcheck=0
```

3. Для обновления системы запустите команду:

```
dnf update
```

Управляющая часть

В кластерном варианте установки узлы управления настраиваются в определенной последовательности:

- Шаг 1: необходимо установить Pacemaker и настроить сервис VIP - виртуального IP-адреса;
- Шаг 2: необходимо установить балансировщик нагрузки (balancer), включить стандартную конфигурацию, а после добавить его как ресурс Pacemaker;
- Шаг 3: необходимо установить и настроить сервис broker;
- Шаг 4: необходимо установить и настроить сервис control.

Каждый шаг должен быть выполнен для всех узлов управления перед переходом к следующему.

Дальнейшие подразделы объяснят подробности выполнения каждого шага.

Установка системы управления кластером Pacemaker

Для установки Pacemaker необходимо выполнить следующее.

1. Установите пакеты Pacemaker:

```
dnf -y install pacemaker corosync pcs
```

2. Создайте пользователя hacluster в ОС узлов и укажите пароль, он необходим для средство аутентификации при сборке кластера Pacemaker:

```
passwd hacluster
```

3. Добавьте разрешение на использование портов Pacemaker в фаерволе:

```
firewall-cmd --permanent --add-service=high-availability
firewall-cmd --add-service=high-availability
```

4. Запустите сервис pcsd и добавьте его в автозапуск:

```
systemctl start pcsd.service
systemctl enable pcsd.service
```

5. В одном из узлов пройдите процесс аутентификации узлов кластера Pacemaker с указанием пароля пользователя hacluster (IP-адреса представлены в виде примера):

```
pcs host auth 192.168.122.33 192.168.122.153 192.168.122.11
```

6. В одном из узлов после успешной аутентификации запустите процесс настройки кластера между узлами (vcorecluster - это имя кластера Pacemaker):

```
pcs cluster setup vcorecluster 192.168.122.33 192.168.122.153 192.168.122.11
```

7. В одном из узлов запустите кластер и включите правила автозапуска кластера при включении узла:

```
pcs cluster start --all
pcs cluster enable --all
```

8. Если не планируется использовать правила fencing, доступные Pacemaker, то в одном из узлов необходимо запустить команду, выключающая проверку настройки механизма STONITH:

```
pcs property set no-quorum-policy=ignore
pcs property set stonith-enabled=false
```

9. В одном из узлов проверьте статус запущенного кластера.

```
pcs status
```

10. В выводе статуса вы должны получить примерно следующее:

```
[root@controller0 /]# pcs status
Cluster name: vcorecluster

Cluster Summary:
  * Stack: corosync
  * Current DC: 192.168.122.11 (version 2.1.4-4.fc35-dc6eb4362e) - partition with
quorum
  * Last updated: Fri Oct 21 14:20:40 2022
  * Last change:  Fri Oct 21 14:19:54 2022 by hacluster via crmd on 192.168.122.11
  * 3 nodes configured
  * 0 resource instances configured

Node List:
  * Online: [ 192.168.122.11 192.168.122.33 192.168.122.153 ]

Full List of Resources:
  * No resources

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

11. Если статусы сервисов не равны Active/enabled, то попробуйте подождать 2-3 минуты, для активации кластера Pacemaker требуется время.

Настройка виртуального IP-адреса

Для входа в кластер vCore требуется использовать единый IP-адрес, который будет настраиваться в одном из интерфейсов управления узлов. Для этого в Pacemaker необходимо создать ресурс, реализующий функцию виртуального IP-адреса. Вам необходимо выбрать незанятый IP-адрес в подсети управления. В качестве примера примем, что таким адресом будет 192.168.122.34, а маска для адреса - /32.

1. В одном из узлов выполните следующую команду:

```
pcs resource create vcore_vip ocf:heartbeat:IPaddr2 ip=192.168.122.34
 cidr_netmask=32 op monitor interval=30s
```

2. В одном из узлов проверьте статус кластера. Вывод должен измениться так:

```
Cluster name: vcorecluster
Cluster Summary:
  * Stack: corosync
  * Current DC: 192.168.122.11 (version 2.1.4-4.fc35-dc6eb4362e) - partition with
quorum
  * Last updated: Fri Oct 21 18:46:43 2022
  * Last change:  Fri Oct 21 14:28:46 2022 by root via cibadmin on 192.168.122.33
  * 3 nodes configured
  * 1 resource instances configured

Node List:
```

```
  * Online: [ 192.168.122.11 192.168.122.33 192.168.122.153 ]

Full List of Resources:
  * vcore_vip   (ocf::heartbeat:IPaddr2):        Started 192.168.122.33
<------ Появился ресурс VIP-адреса.

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

3. В нашем примере видно, что VIP-адрес установлен для интерфейса в узле с адресом 192.168.122.33. Чтобы убедиться в этом, достаточно получить список интерфейсов командой `ip a` в узле, где запущен VIP:

```
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 52:54:00:52:a8:47 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.33/24 brd 192.168.122.255 scope global dynamic noprefixroute
enp1s0
       valid_lft 2376sec preferred_lft 2376sec
    inet 192.168.122.34/32 scope global enp1s0        <------- Виртуальный адрес.
       valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe52:a847/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Установка балансировщика нагрузки

Установки балансера выполняется в три этапа:

1. Установить пакет balancer и произвести первоначальную настройку;
2. Необходимо создать ресурс systemd в Pacemaker в режиме Active/Standby, который будет запускать balancer в одном узле из имеющихся;
3. Необходимо указать правило, что balancer должен всегда запускаться вместе с виртуальным адресом.

Установка сервиса balancer

Для установки и настройки balancer требуется сделать следующее:

1. Установите пакет balancer:

```
dnf -y install vcore-balancer
```

2. В файле `/etc/sysconfig/vcore-balancer` укажите адрес прослушивания для сервиса. Для первого параметра необходимо указать выбранный VIP-адрес, а для второго - реальный IP-адрес интерфейса:

```
BALANCER_BIND_ADDR="192.168.122.34"
BALANCER_PRIVATE_BIND_ADDR="192.168.122.33"
```

3. Очистите файл `/opt/vcore/balancer/config/standalone.xml`:

```
> /opt/vcore/balancer/config/standalone.xml
```

4. Включите следующую конфигурацию в файл `/opt/vcore/balancer/config/standalone.xml`:

/opt/vcore/balancer/config/standalone.xml

```
<?xml version='1.0' encoding='UTF-8'?>

<server xmlns="urn:jboss:domain:16.0">
    <extensions>
        <extension module="org.wildfly.extension.elytron"/>
        <extension module="org.wildfly.extension.io"/>
```

```
        <extension module="org.wildfly.extension.undertow"/>
    </extensions>
    <management>
        <identity security-domain="ManagementDomain"/>
        <audit-log>
            <formatters>
                <json-formatter name="json-formatter"/>
            </formatters>
            <handlers>
                <file-handler name="file" formatter="json-formatter" path="audit-
log.log" relative-to="jboss.server.data.dir"/>
            </handlers>
            <logger log-boot="true" log-read-only="false" enabled="false">
                <handlers>
                    <handler name="file"/>
                </handlers>
            </logger>
        </audit-log>
        <management-interfaces>
            <http-interface http-authentication-factory="management-http-
authentication">
                <http-upgrade enabled="true" sasl-authentication-factory="manageme
nt-sasl-authentication"/>
                <socket-binding http="management-http"/>
            </http-interface>
        </management-interfaces>
        <access-control provider="simple">
            <role-mapping>
                <role name="SuperUser">
                    <include>
                        <user name="$local"/>
                    </include>
                </role>
            </role-mapping>
        </access-control>
    </management>
    <profile>
        <subsystem xmlns="urn:wildfly:elytron:13.0" final-providers="combined-
providers" disallowed-providers="OracleUcrypto">
            <providers>
                <aggregate-providers name="combined-providers">
                    <providers name="elytron"/>
                    <providers name="openssl"/>
                </aggregate-providers>
                <provider-loader name="elytron" module="org.wildfly.security.elytr
on"/>
                <provider-loader name="openssl" module="org.wildfly.openssl"/>
            </providers>
            <audit-logging>
                <file-audit-log name="local-audit" path="audit.log" relative-to="j
boss.server.log.dir" format="JSON"/>
            </audit-logging>
            <security-domains>
                <security-domain name="ApplicationDomain" default-realm="Applicati
onRealm" permission-mapper="default-permission-mapper">
                    <realm name="ApplicationRealm" role-decoder="groups-to-roles"/
>
                    <realm name="local"/>
                </security-domain>
                <security-domain name="ManagementDomain" default-realm="Management
Realm" permission-mapper="default-permission-mapper">
                    <realm name="ManagementRealm" role-decoder="groups-to-roles"/>
                    <realm name="local" role-mapper="super-user-mapper"/>
                </security-domain>
            </security-domains>
            <security-realms>
                <identity-realm name="local" identity="$local"/>
                <properties-realm name="ApplicationRealm">
```

```
                    <users-properties path="application-users.properties"
relative-to="jboss.server.config.dir" digest-realm-name="ApplicationRealm"/>
                    <groups-properties path="application-roles.properties"
relative-to="jboss.server.config.dir"/>
                </properties-realm>
                <properties-realm name="ManagementRealm">
                    <users-properties path="mgmt-users.properties" relative-to="jb
oss.server.config.dir" digest-realm-name="ManagementRealm"/>
                    <groups-properties path="mgmt-groups.properties" relative-to="
jboss.server.config.dir"/>
                </properties-realm>
            </security-realms>
            <mappers>
                <simple-permission-mapper name="default-permission-mapper"
mapping-mode="first">
                    <permission-mapping>
                        <principal name="anonymous"/>
                        <permission-set name="default-permissions"/>
                    </permission-mapping>
                    <permission-mapping match-all="true">
                        <permission-set name="login-permission"/>
                        <permission-set name="default-permissions"/>
                    </permission-mapping>
                </simple-permission-mapper>
                <constant-realm-mapper name="local" realm-name="local"/>
                <simple-role-decoder name="groups-to-roles" attribute="groups"/>
                <constant-role-mapper name="super-user-mapper">
                    <role name="SuperUser"/>
                </constant-role-mapper>
            </mappers>
            <permission-sets>
                <permission-set name="login-permission">
                    <permission class-name="org.wildfly.security.auth.permission.L
oginPermission"/>
                </permission-set>
                <permission-set name="default-permissions"/>
            </permission-sets>
            <http>
                <http-authentication-factory name="management-http-authentication"
 security-domain="ManagementDomain" http-server-mechanism-factory="global">
                    <mechanism-configuration>
                        <mechanism mechanism-name="DIGEST">
                            <mechanism-realm realm-name="ManagementRealm"/>
                        </mechanism>
                    </mechanism-configuration>
                </http-authentication-factory>
                <provider-http-server-mechanism-factory name="global"/>
            </http>
            <sasl>
                <sasl-authentication-factory name="application-sasl-
authentication" sasl-server-factory="configured" security-domain="ApplicationDomai
n">
                    <mechanism-configuration>
                        <mechanism mechanism-name="JBOSS-LOCAL-USER" realm-mapper="
local"/>
                        <mechanism mechanism-name="DIGEST-MD5">
                            <mechanism-realm realm-name="ApplicationRealm"/>
                        </mechanism>
                    </mechanism-configuration>
                </sasl-authentication-factory>
                <sasl-authentication-factory name="management-sasl-authentication"
 sasl-server-factory="configured" security-domain="ManagementDomain">
                    <mechanism-configuration>
                        <mechanism mechanism-name="JBOSS-LOCAL-USER" realm-mapper="
local"/>
                        <mechanism mechanism-name="DIGEST-MD5">
                            <mechanism-realm realm-name="ManagementRealm"/>
                        </mechanism>
```

```
                    </mechanism-configuration>
                </sasl-authentication-factory>
                <configurable-sasl-server-factory name="configured" sasl-server-
factory="elytron">
                    <properties>
                        <property name="wildfly.sasl.local-user.default-user"
value="$local"/>
                    </properties>
                </configurable-sasl-server-factory>
                <mechanism-provider-filtering-sasl-server-factory name="elytron"
sasl-server-factory="global">
                    <filters>
                        <filter provider-name="WildFlyElytron"/>
                    </filters>
                </mechanism-provider-filtering-sasl-server-factory>
                <provider-sasl-server-factory name="global"/>
            </sasl>
            <tls>
                <key-stores>
                    <key-store name="applicationKS">
                        <credential-reference clear-text="password"/>
                        <implementation type="JKS"/>
                        <file path="application.keystore" relative-to="jboss.serve
r.config.dir"/>
                    </key-store>
                </key-stores>
                <key-managers>
                    <key-manager name="applicationKM" key-store="applicationKS"
generate-self-signed-certificate-host="localhost">
                        <credential-reference clear-text="password"/>
                    </key-manager>
                </key-managers>
                <server-ssl-contexts>
                    <server-ssl-context name="applicationSSC" key-manager="applica
tionKM"/>
                </server-ssl-contexts>
            </tls>
        </subsystem>
        <subsystem xmlns="urn:jboss:domain:io:3.0">
            <worker name="default"/>
            <buffer-pool name="default"/>
        </subsystem>
        <subsystem xmlns="urn:jboss:domain:undertow:12.0" default-server="default-
server" default-servlet-container="default" statistics-enabled="$
{wildfly.undertow.statistics-enabled:${wildfly.statistics-enabled:false}}">
            <buffer-cache name="default"/>
            <server name="default-server">
                <http-listener name="default" socket-binding="http" redirect-
socket="https" enable-http2="true" proxy-address-forwarding="true" max-post-size="
10737418240"/>
                <http-listener name="management" socket-binding="mcmp-management"
enable-http2="true"/>
                <host name="default-host" alias="localhost">
                    <filter-ref name="load-balancer"/>
                </host>
            </server>
            <server name="default-server-broker">
                <http-listener name="default-broker" socket-binding="http-broker"
enable-http2="false"/>
                <http-listener name="management-broker" socket-binding="mcmp-
management-broker" enable-http2="false"/>
                <host name="default-host" alias="localhost">
                    <filter-ref name="load-balancer-broker"/>
                </host>
            </server>
            <servlet-container name="default"/>
            <filters>
```

```
                <mod-cluster name="load-balancer" management-socket-binding="mcmp-
management" advertise-socket-binding="modcluster" enable-http2="false" max-retries="
3">
                    <single-affinity/>
                </mod-cluster>
                <mod-cluster name="load-balancer-broker" management-socket-binding="
mcmp-management-broker" advertise-socket-binding="modcluster-broker" enable-http2="
false" max-retries="3">
                 <single-affinity/>
                </mod-cluster>
            </filters>
        </subsystem>
    </profile>
    <interfaces>
        <interface name="management">
            <inet-address value="${jboss.bind.address.private:127.0.0.1}"/>
        </interface>
        <interface name="private">
            <inet-address value="${jboss.bind.address.private:127.0.0.1}"/>
        </interface>
        <interface name="public">
            <inet-address value="${jboss.bind.address:127.0.0.1}"/>
        </interface>
    </interfaces>
    <socket-binding-group name="standard-sockets" default-interface="public" port-
offset="${jboss.socket.binding.port-offset:0}">
        <socket-binding name="http" port="${jboss.http.port:8080}"/>
        <socket-binding name="https" port="${jboss.https.port:8443}"/>
        <socket-binding name="management-http" interface="management" port="$
{jboss.management.http.port:9990}"/>
        <socket-binding name="management-https" interface="management" port="$
{jboss.management.https.port:9993}"/>
        <socket-binding name="mcmp-management" interface="private" port="$
{jboss.mcmp.port:8090}"/>
        <socket-binding name="modcluster" interface="private" multicast-address="$
{jboss.modcluster.multicast.address:224.1.0.105}" multicast-port="23364"/>
        <socket-binding name="http-broker" port="8081"/>
        <socket-binding name="mcmp-management-broker" interface="private" port="80
91"/>
        <socket-binding name="modcluster-broker" interface="private" multicast-
address="224.1.0.105" multicast-port="23365"/>
    </socket-binding-group>
</server>
```

5. Убедитесь, что сервис balancer в systemd не содержит флага автозапуска:

```
systemctl disable vcore-balancer
```

Настройка ресурса balancer

1. Для добавления ресурса balancer в одном из узлов необходимо выполнить следующую команду:

```
pcs resource create vcore_balancer systemd:vcore-balancer op monitor
interval="10s" timeout="15s"
```

2. Проверьте статус кластера. Вывод должен измениться так:

```
Cluster name: vcorecluster
Cluster Summary:
  * Stack: corosync
  * Current DC: 192.168.122.11 (version 2.1.4-4.fc35-dc6eb4362e) - partition with
quorum
  * Last updated: Fri Oct 21 18:53:47 2022
  * Last change:  Fri Oct 21 14:28:46 2022 by root via cibadmin on 192.168.122.33
  * 3 nodes configured
  * 2 resource instances configured

Node List:
```

```
  * Online: [ 192.168.122.11 192.168.122.33 192.168.122.153 ]

Full List of Resources:
  * vcore_vip   (ocf::heartbeat:IPaddr2):        Started 192.168.122.33
  * vcore_balancer      (systemd:vcore-balancer):   Started 192.168.122.153
 <------ Появился ресурс сервиса balancer.

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

3. В выводе можно заметить, что balancer запустился в узле, отличным от того, где настроен VIP-адрес. Для исправления этой ситуации необходимо в одном из узлов запустить команду, который добавит ограничение для ресурсов (colocation):

```
pcs constraint colocation add vcore_balancer with vcore_vip
```

4. Также необходимо включить правило очередности запуска ресурсов: вначале должен запускаться vip-адрес, затем ресурс балансировщика. Для этого в одном из узлов выполните следующую команду:

```
pcs constraint order start vcore_vip then start vcore_balancer
```

5. Ещё раз проверьте статус сервисов:

```
Cluster name: vcorecluster
Cluster Summary:
  * Stack: corosync
  * Current DC: 192.168.122.11 (version 2.1.4-4.fc35-dc6eb4362e) - partition with
quorum
  * Last updated: Fri Oct 21 19:06:44 2022
  * Last change:  Fri Oct 21 14:28:46 2022 by root via cibadmin on 192.168.122.33
  * 3 nodes configured
  * 2 resource instances configured

Node List:
  * Online: [ 192.168.122.11 192.168.122.33 192.168.122.153 ]

Full List of Resources:
  * vcore_vip   (ocf::heartbeat:IPaddr2):        Started 192.168.122.33
  * vcore_balancer      (systemd:vcore-balancer):   Started 192.168.122.33
<------ Адрес равен адресу узла, где запущен VIP.

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

⚠ Pacemaker может запустить balancer на узле без VIP до настройки привязки его ресурса к VIP. В этом случае в статусе будет выводиться лог неудачного запуска balancer:

```
   * vcore_balancer_monitor_10000 on 192.168.122.153 'not running' (7):
call=11, status='complete', exitreason='failed', last-rc-change='Thu Oct 27
17:46:34 2022', queued=0ms, exec=0ms
```

После указания привязки balancer успешно запустится. Для очистки от данных прошлых неудачных запусков запустите команду:

```
pcs resource cleanup vcore_balancer
```

Установка и настройка брокера

Для установки и настройки брокера необходимо выполнить следующее:

1. Установите пакет брокера:

```
dnf -y install vcore-broker
```

2. Брокер будет установлен в каталог `/opt/vcore/broker/`. Откройте файл конфигурации брокера по пути `/opt/vcore/broker/config/application.properties` и включите следующие изменения (в настройках с IP-адресом указывайте *реальный в интерфейсе управления узла*):
   a. настройте адрес прослушивания для брокера
   ```
   quarkus.http.host=192.168.122.33
   ```
   b. Включите поддержку балансировщика нагрузки (предоставляет функцию автоматической регистрации сервиса в балансировщике):
   ```
   quarkus.mod-cluster-client.enabled=true
   quarkus.mod-cluster-client.services.api.host=192.168.122.33
   quarkus.mod-cluster-client.services.api.port=8081
   quarkus.mod-cluster-client.advertise-multicast-port=23365
   quarkus.mod-cluster-client.services.api.contexts=/api/storage, /api/transport
   ```
   c. Настройте адреса прослушивания для базы данных и включите обанружение экземпляров кластера базы данных через multicast:
   ```
   provider.storage.use-multicast-discovery=true
   provider.storage.multicast-interface-addresses=192.168.122.33
   provider.storage.bind-address=192.168.122.33
   provider.storage.jdbc-address=192.168.122.33
   ```
   d. Так же включите обнаружение брокера через multicast для агентов вычислительных узлов и control, а также адрес прослушивания для сервиса сообщений:
   ```
   provider.transport.use-multicast-discovery=true
   provider.transport.multicast-interface-address=192.168.122.33
   ```
   e. Укажите *уникальный для каждого узла* ID экземпляра кластера базы данных в формате UUID4:
   ```
   provider.storage.consistent-id=d519d321-931d-4f3d-b7f5-30c9c4180640
   ```
   f. При необходимости усиления безопасности доступа к брокеру измените параметры входа для экземпляров кластера и control/агентов вычисления (пароль должен быть закодирован в base64):
   ```
   # Broker Cluster Auth Data
   provider.transport.cluster.user=user
   provider.transport.cluster.password=Q0hBTkdFTUUhIQ==
   # Broker Client (control/agent) Auth Data
   provider.transport.user=user
   provider.transport.password=Q0hBTkdFTUUhIQ==
   ```
   g. В одном из узлов включите возможность миграции базы данных:
   ```
   provider.migration.enabled=true
   ```

   Важно

   При запуске сервиса в сетях с Dual Stack необходимо использовать IPv4-адреса в конфигурации.

3. После настройки брокера на всех узлах необходимо строго последовательно (с ожиданием возвращения ввода в терминале) запустить каждый сервис брокера:
   ```
   systemctl start vcore-broker
   ```
4. При первой инициализации очерёдность запуска сервисов брокера не важна, однако она будет иметь значение при обновлении платформы (подробно описано в статье про обновление платформы). Также брокер не нужно добавлять как ресурс в Pacemaker.

Установка и настройка контрола

Для установки и настройки control необходимо выполнить следующее:

1. Установите пакет control:

```
dnf -y install vcore-control
```

2. control будет установлен в каталог `/opt/vcore/control/`. Откройте файл конфигурации control по пути `/opt/vcore/control/config/application.properties` и включите следующие изменения:
   a. настройте адрес прослушивания для брокера (нужно указать *реальный IP-адрес* в интерфейсе управления узла*)*

```
quarkus.http.host=192.168.122.33
```

   b. Укажите адреса доступных экземпляров баз данных (первый адрес всегда должен быть равен реальному IP-адресу узла, где применяется конфигурация):

```
quarkus.datasource.jdbc.url=jdbc:ignite:thin://
192.168.122.33,192.168.122.153,192.168.122.11?distributedJoins=true
```

   c. Укажите адрес брокера, доступный через балансировщик нагрузки:

```
broker-rest-api/mp-rest/url=http://192.168.122.34:8081/api
```

   d. Укажите адрес сервиса сообщений. Обязательно нужно указать multicast-группу 231.7.7.7 и протокол UDP:

```
quarkus.artemis.url=udp://231.7.7.7:9876?
clientFailureCheckPeriod=5000&retryInterval=1000&reconnectAttempts=5
```

   e. Укажите *уникальный для каждого узла* идентификатор в формате UUID4:

```
control.node-id=da1c181a-2921-4be0-a21e-ca15081a20d8
```

   f. Включите поддержку регистрации в балансировщике нагрузки:

```
quarkus.mod-cluster-client.enabled=true
quarkus.mod-cluster-client.advertise-multicast-port=23364
```

   g. Если в настройках брокера вы изменили логин и пароль для transport.user, то укажите их:

```
quarkus.artemis.username=user
quarkus.artemis.password=Q0hBTkdFTUUhIQ==
```

> **Важно**
>
> При запуске сервиса в сетях с Dual Stack необходимо использовать IPv4-адреса в конфигурации.

3. После настройки control на всех узлах запустите сервисы:

```
systemctl start vcore-control
```

4. Попробуйте подключиться к веб-интерфейсу, используя VIP-адрес и порт 8080:

```
http://192.168.122.34:8080
```

> **Примечание**
>
> При использовании FirewallD в качестве системы управления Firewall необходимо открыть порты сервисов. Для этого используйте следующие команды:

```
firewall-cmd --zone=public --add-service=vcore-broker --permanent
firewall-cmd --zone=public --add-service=vcore-control --permanent
firewall-cmd --zone=public --add-service=vcore-balancer --permanent
```

Перезапустите Firewall:

```
firewall-cmd --reload
```

Для проверки портов выполните:

```
nc -v IP_ADDR PORT
```

Где:

- IP_ADDR - IP-адрес узла;
- PORT - Порт сервиса.

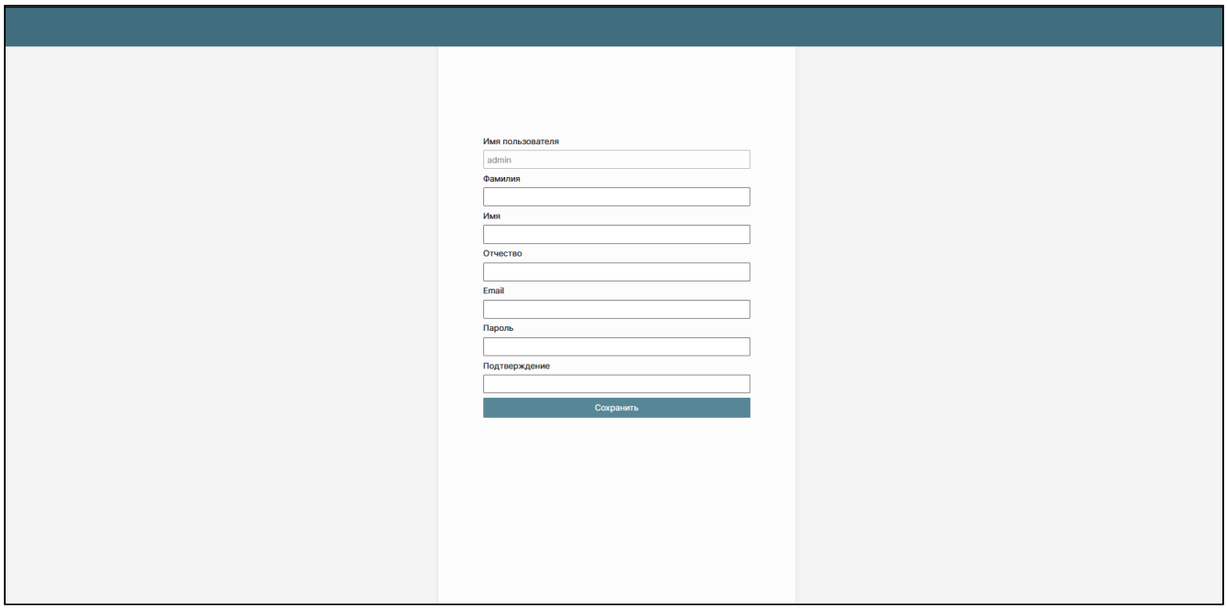
Список портов сервисов по умолчанию доступен по ссылке (см. стр. 51).

Настройка панели управления

После запуска Control по адресу http://VIP_DNS_ADDR:8080 (в нашем примере - это адрес 192.168.122.134) будет доступен веб-интерфейс платформы. Укажите первичные данные для пользователя admin:

Информация

Для корректной работы системы рекомендуется использовать Google Chrome или Firefox актуальных версий.



Форма создания пользователя

Заполните поля формы создания пользователя. Подтвердите создание кнопкой "Сохранить". После этого укажите следующие данные для входа в систему:

- домен: master;
- пользователь: admin;
- пароль: указанный для пользователя admin.

Вычислительная часть

Процесс установки и настройки вычислительной части не отличается от простой инсталляции. Данная процедура подробно описана в разделе - "Установка гипервизора" (см. стр. 12).

Важно

После установки и настройки на вычислительном узле ПО Базис узел автоматически подключится к платформе со статусом "Обслуживание". Для ввода узла в эксплуатацию с таким состоянием его необходимо активировать, добавив его в один из кластеров. Данная процедура подробна описана в руководстве Администратора vCore.

Интеграция Базис.vCore v.1.3 с Базис.Virtual Security

- Введение (см. стр. 33)
- Управляющая часть (см. стр. 33)
- Вычислительная часть (см. стр. 34)

Введение

Данная статья описывает процесс настройки платформы Базис.vCore для корректной работы с Базис.Virtual Security в рамках единого кластера управления.

Для установки vCore используйте один из возможных вариантов:

- Демонстрационная инсталляция[4];
- Простая инсталляция[5];
- Инсталляция с поддержкой высокой доступности[6].

Подробный процесс установки Базис.Virtual Security описан в "Руководстве по установке" данного продукта.

Управляющая часть

Настройте конфигурационный файл сервиса Control - `/opt/vcore/control/config/application.properties:`

```
quarkus.http.port=8086
broker-rest-api/mp-rest/url=http://IPADDRESS:8085/api
```

Настройте конфигурационный файл сервиса Broker - `/opt/vcore/broker/config/application.properties:`

```
quarkus.http.port=8085
quarkus.mod-cluster-client.services.api.port=8085
```

Настройте firewall:

```
firewall-cmd --permanent --add-port=8085/tcp
firewall-cmd --permanent --add-port=8086/tcp
firewall-cmd --reload
```

Перезапустите сервисы Control и Broker:

```
systemctl restart tionix-vcore-broker vcore-control
```

После перезапуска сервисов веб-интерфейс платформы vCore будет доступен по адресу - `http://IP-ADDRESS:8086`.

Добавьте репозиторий проекта в - `/etc/yum.repos.d/bvs.repo`:

```
[bvs]
name=BVS $releasever
baseurl=https://tvs-public:tvs-public@maven.tionix.ru/artifactory/tvs-rpm/dev/fedora35/
$basearch
enabled=1
gpgcheck=0

[bvs-noarch]
name=BVS $releasever
baseurl=https://tvs-public:tvs-public@maven.tionix.ru/artifactory/tvs-rpm/dev/fedora35/
noarch
enabled=1
```

4 https://conf.tionix.ru/pages/viewpage.action?pageId=34884942O&src=contextnavpagetreemode
5 https://conf.tionix.ru/pages/viewpage.action?pageId=348849423&src=contextnavpagetreemode
6 https://conf.tionix.ru/pages/viewpage.action?pageId=34884944O&src=contextnavpagetreemode

```
gpgcheck=0

[vcore-contrib]
name=vCore Contrib Packages
baseurl=https://vcore-public:vcore-public@maven.tionix.ru/artifactory/vcore-contrib/
fedora/linux/$releasever/$basearch
enabled=1
gpgcheck=0
```

Установите пакет postgresql:

```
dnf -y install postgresql12-server
```

Установите пакет tionix-tvs-system:

```
dnf install tionix-tvs-system
```

Вычислительная часть

В данном разделе описана процедура установки агентского программного обеспечения Базис.Virtual Security на вычислительном узле (гипервизоре) Базис.vCore.

Добавьте репозиторий проекта в - `/etc/yum.repos.d/bvs.repo`:

```
[bvs]
name=BVS $releasever
baseurl=https://tvs-public:tvs-public@maven.tionix.ru/artifactory/tvs-rpm/dev/
fedora35/$basearch
enabled=1
gpgcheck=0

[bvs-noarch]
name=BVS $releasever
baseurl=https://tvs-public:tvs-public@maven.tionix.ru/artifactory/tvs-rpm/dev/fedora35/
noarch
enabled=1
gpgcheck=0
```

Установите необходимые пакеты:

```
dnf install -y tionix-tvs-agent
```

Перезапустите службу агента вычислительного узла:

```
systemctl restart vcore-agent
```

Обновление Базис.vCore v.1.3

Простая инсталляция Базис.vCore v1.3

Предварительные действия перед обновлением

На период обновления платформы необходимо остановить все работы в облаке:

1. Остановите ВМ запущенные ВМ.
2. Переведите гипервизоры в режим обслуживания.
3. Выключите все сервисы агентов, подключенные к системе управления vCore.

Управляющая часть

Обновление пакетов

Остановите службы системы управления:

```
systemctl stop vcore-control
systemctl stop vcore-broker
```

Проверьте наличие обновлений для пакетов vCore:

```
dnf check-update --refresh | grep 'vcore-'
```

Обновите пакеты Базис vCore:

```
dnf -y update vcore-control vcore-broker
```

Для обновления всей системы используйте обычную команду update:

```
dnf -y update
```

Важно

При перезапуске узла при обновлении системных пакетов (например, ядра) необходимо предварительно выключить автозагрузку служб Базис vCore:

```
systemctl disable vcore-control vcore-broker
```

После чего произведите перезапуск.

Обновление конфигурации сервисов

После обновления пакетов имеющийся файл конфигурации сервисов будут переименован как `applications.properties.rpmsave`.

```
ls /opt/vcore/broker/config/
application.properties  application.properties.rpmsave

ls /opt/vcore/control/config/
application.properties  application.properties.rpmsave
```

В новом файле `application.properties` необходимо перенести все указанные в старой конфигурации параметры сервисов. В большинстве случаев необходимо перенести настройку следующих параметров:

- для broker:

```
quarkus.http.host
provider.storage.multicast-interface-addresses
provider.storage.static-discovery-addresses
```

```
provider.storage.jdbc-address
provider.storage.use-multicast-discovery
provider.transport.use-multicast-discovery
provider.transport.multicast-interface-address
provider.transport.static-discovery-addresses
provider.transport.bind-address
provider.transport.user
provider.transport.password
```

- для control:

```
quarkus.http.host
control.node-id
quarkus.datasource.jdbc.url
broker-rest-api/mp-rest/url
quarkus.artemis.url
quarkus.artemis.username
quarkus.artemis.password
```

Сохраните изменения.

Запуск сервисов

Запустите службы:

```
systemctl start vcore-broker
systemctl start vcore-control
```

Проверьте, что по порту 8082 через браузер vCore загружает веб-интерфейс.

> ⚠ Если веб-интерфейс недоступен по порту 8082, то, скорее всего, после обновления не добавилось правило на разрешение к подключению на этот порт в firewalld. В этом случае запустите команды:
>
> ```
> firewall-cmd --add-port=8082/tcp  --permanent
> firewall-cmd --reload
> ```
>
> И попробуйте снова загрузить веб-интерфейс через браузер.

Включите автозагрузку служб:

```
systemctl enable vcore-control vcore-broker
```

Вычислительная часть

Обновление агента и конфигуратора

Остановите службу агента вычислительного узла:

```
systemctl stop vcore-agent
```

Проверьте наличие обновлений для пакетов vCore:

```
dnf check-update --refresh | grep 'vcore-'
```

Обновите пакеты Базис vCore:

```
dnf update -y vcore-agent vcore-hyper-configurator
```

Для обновления всей системы используйте обычную команду update:

```
dnf -y update
```

Важно

При перезапуске узла при обновлении системных пакетов (например, ядра) необходимо предварительно выключить автозагрузку служб Базис vCore:

```
systemctl disable vcore-agent
```

После чего произведите перезапуск.

Обновление конфигурации агента

После обновления пакетов имеющийся файл конфигурации агента будут переименован как `applications.properties.rpmsave`.

```
ls /opt/vcore/agent/config/
application.properties  application.properties.rpmsave
```

В новом файле `application.properties` необходимо перенести все указанные в старой конфигурации параметры агента. В большинстве случаев необходимо перенести настройку следующих параметров:

```
quarkus.http.host
agent.realm-id
agent.dc-id
agent.node-id
quarkus.artemis.url
quarkus.artemis.username
quarkus.artemis.password
```

Сохраните изменения.

Запуск агента

Запустите службу агента:

```
systemctl start vcore-agent
```

В "Инфраструктура — Узлы" веб-интерфейса vCore убедитесь, что агент перешёл в статус "Активный", а версия агента равна 1.1.0.

Если агент не запускается с ошибкой NmstateError, то перед запуском агента перезапустите сервис NetworkManager:

```
systemctl restart NetworkManager
```

После этого произведите запуск службы агента.

Включите автозагрузку службы агента:

```
systemctl enable vcore-agent
```

Инсталляция Базис.vCore v.1.3 с поддержкой высокой доступности

- Предварительные действия перед обновлением (см. стр. 38)
- Управляющая часть (см. стр. 38)
  - Выключение сервисов (см. стр. 38)
  - Установка новых компонентов (см. стр. 38)
  - Обновление пакетов (см. стр. 38)
  - Обновление конфигурации сервисов (см. стр. 39)
    - Сервисы control и broker (см. стр. 39)
- Обновление вычислительной части (см. стр. 40)
  - Обновление агента и конфигуратора гипервизора (см. стр. 40)

- Обновление конфигурации агента (см. стр. 41)
- Запуск агента (см. стр. 41)

Предварительные действия перед обновлением

На период обновления платформы необходимо остановить все работы в облаке:

1. Остановите ВМ запущенные ВМ.
2. Переведите гипервизоры в режим обслуживания.
3. Выключите все сервисы агентов, подключенные к системе управления vCore.

Управляющая часть

Выключение сервисов

При обновлении управляющей части необходимо строго соблюдать порядок работ на узлах.

Остановите службы системы управления во всех узлах с учётом статуса миграции (статус миграции необходимо проверять в параметре provider.migration.enabled, который указан в файле `/opt/vcore/{control,broker}/config/application.properties`):

- вначале control последовательно на всех узлах:

```
systemctl stop vcore-control
```

- затем broker последовательно на всех узлах. В первую очередь необходимо выключить брокер с включенной миграцией (true), а потом - брокеры с выключённой (false):

```
systemctl stop vcore-broker
```

До окончания процесса обновления выключите автозапуск этих сервисов:

```
systemctl disable vcore-control vcore-broker
```

Установка новых компонентов

В версии 1.1 в составе узлов управления появились новые компоненты: система управления Pacemaker, виртуальный IP-адрес и балансировщик нагрузки balancer. Их установка и настройка подробно описана в инструкции по установке vCore:

- Установка Pacemaker (см. стр. 22);
- Настройка виртуального IP-адреса (см. стр. 23);
- Настройка балансировщика нагрузки (см. стр. 24).

После установки этих компонентов можно переходить к обновлению control и broker.

Обновление пакетов

Проверьте наличие обновлений для пакетов vCore:

```
dnf check-update --refresh | grep 'vcore-'
```

Обновите пакеты Базис vCore:

```
dnf -y update vcore-control vcore-broker
```

Включите автозагрузку служб:

```
systemctl enable vcore-control vcore-broker
```

В журнале balancer можно увидеть данные регистрации сервисов control и broker.

Обновление конфигурации сервисов

Сервисы control и broker

После обновления пакетов имеющийся файл конфигурации для сервисов control и broker будут переименован как `applications.properties.rpmsave.`

Важно

Файлы `applications.properties.rpmsave` появляются только при:

- увеличении мажорной версии: 1.0 - 1.1, 1.1 - 2.0;
- если файл отличается от файла в пакете.

```
ls /opt/vcore/broker/config/
application.properties  application.properties.rpmsave

ls /opt/vcore/control/config/
application.properties  application.properties.rpmsave
```

Информация

Для применения старой конфигурации можно воспользоваться данной командой:

```
mv /opt/vcore/broker/config/application.properties.rpmsave /opt/vcore/broker/
config/application.properties
mv /opt/vcore/control/config/application.properties.rpmsave /opt/vcore/control/
config/application.properties
```

В новом файле `application.properties` необходимо перенести все указанные в старой конфигурации параметры сервисов. В большинстве случаев необходимо перенести настройку следующих параметров:

- для broker:

```
quarkus.http.host
provider.storage.multicast-interface-addresses
provider.storage.bind-address
provider.storage.jdbc-address
provider.transport.multicast-interface-address
provider.storage.consistent-id
provider.transport.cluster.user
provider.transport.cluster.password
provider.transport.user
provider.transport.password
provider.migration.enabled
```

- для control:

```
quarkus.http.host
quarkus.datasource.jdbc.url
quarkus.artemis.url
control.node-id
quarkus.artemis.username
quarkus.artemis.password
```

В адресе quarkus.artemis.url должен быть указан URL с протоколом UDP и адресом multicast 231.7.7.7:9876

В качестве адреса балансера укажите виртуальный IP-адрес:

```
broker-rest-api/mp-rest/url=http://192.168.122.34:8081/api
```

В параметрах control и broker включите поддержку работы с балансером (по умолчанию в версии 1.0 он выключен). В качестве адреса укажите реальный IP-адрес узла:

- broker:

```
quarkus.mod-cluster-client.enabled=true
quarkus.mod-cluster-client.services.api.host=192.168.122.33
quarkus.mod-cluster-client.services.api.port=8081
quarkus.mod-cluster-client.advertise-multicast-port=23365
quarkus.mod-cluster-client.services.api.contexts=/api/storage, /api/transport
```

- control:

```
quarkus.mod-cluster-client.enabled=true
quarkus.mod-cluster-client.advertise-multicast-port=23364
```

Убедитесь, что следующие параметры в сервисе broker имеют значение true:

```
provider.storage.use-multicast-discovery=true
provider.transport.use-multicast-discovery=true
```

Сохраните изменения.

Запустите службы на всех узлах кластера в обратном порядке (статус миграции необходимо проверять в параметре provider.migration.enabled, который указан в файле `/opt/vcore/{control,broker}/config/application.properties`):

- вначале broker последовательно на всех узлах. В первую очередь необходимо включить брокеры с выключенной миграцией (false), и лишь потом в конце - брокер с включённой (true):

```
systemctl start vcore-broker
```

- затем control последовательно на всех узлах:

```
systemctl start vcore-control
```

Убедитесь, что сервисы успешно запустились.

Важно

После обновления платформы может возникнуть ошибка:

```
Failed to execute query because cache partition has been lostPart
```

Для ее исправления выполните REST запрос:

```
curl -XPUT -H "Content-Type: application/json" http://broker-addr:8081/api/storage/manage/resetLostPartitions
```

Где: broker-addr - IP-адрес узла брокера.

Обновление вычислительной части

Обновление агента и конфигуратора гипервизора

Остановите службу агента:

```
systemctl stop vcore-agent
```

Проверьте наличие обновлений для пакетов vCore:

```
dnf check-update --refresh | grep 'vcore-'
```

Обновите пакеты Базис vCore:

```
dnf update -y vcore-agent vcore-hyper-configurator
```

Для обновления всей системы используйте обычную команду update:

```
dnf -y update
```

Важно

При перезапуске узла при обновлении системных пакетов (например, ядра) необходимо предварительно выключить автозагрузку служб Базис vCore:

```
systemctl disable vcore-agent
```

После чего произведите перезапуск.

Обновление конфигурации агента

После обновления пакетов имеющийся файл конфигурации агента будут переименован как `applications.properties.rpmsave`.

```
ls /opt/vcore/agent/config/
application.properties  application.properties.rpmsave
```

В новом файле `application.properties` необходимо перенести все указанные в старой конфигурации параметры агента. В большинстве случаев необходимо перенести настройку следующих параметров:

```
quarkus.http.host
agent.realm-id
agent.dc-id
agent.node-id
quarkus.artemis.url
quarkus.artemis.username
quarkus.artemis.password
```

В адресе quarkus.artemis.url должен быть указан URL с протоколом UDP и адресом multicast 231.7.7.7:9876

Сохраните изменения.

Запуск агента

Запустите службу агента:

```
systemctl start vcore-agent
```

В "Инфраструктура — Узлы" интерфейса vCore убедитесь, что агент перешёл в статус "Активный", а версия агента равна 1.1.0.

Если агент не запускается с ошибкой NmstateError, то перед запуском агента перезапустите сервис NetworkManager:

```
systemctl restart NetworkManager
```

После этого произведите запуск службы агента.

Включите автозагрузку службы агента:

```
systemctl enable vcore-agent
```

⚠ Обратите внимание, что на данный момент обновление гипервизора через пакетный менеджер не поднимает базовую версию гипервизора.

Обновление vCore 1.2 до 1.3

Введение

Из-за ребрендинга пакетов vCore, при котором с дистрибутива удалялись все упоминания прошлого наименования проекта (TVC), изменились ключевые параметры и конфигурация пакетов, а так же сами названия частей продукта. Поэтому обычное обновление пакетов не будет работать должным образом, нужно ручное вмешательство. Процесс обновления будет состоять из нескольких шагов:

1. Выключение всех сервисов vCore (в узлах контроллера и в агентах ВУ). При наличии кластера управления - дополнительно выключить ресурс балансера в Pacemaker и сохранение конфигурации balancer.
2. Создание копии конфигурации vCore внутри /opt/tvc в отдельный каталог на всех узлах без исключений.
3. Удаление пакетов с именованием tvc.
4. Установка пакетов с именованием vcore.
5. Возвращение копии конфигурации vCore в каталог /opt/vcore
6. Запуск всех сервисов.

Шаги обновления

- Выключение объектов в vCore (см. стр. 42)
- Выключение всех сервисов vCore (см. стр. 42)
- Создание копии конфигурации (см. стр. 43)
- Удаление пакетов с именем tvc (см. стр. 43)
- Установка новых пакетов (см. стр. 43)
- Возвращение конфигурации vCore (см. стр. 46)
- Запуск сервисов (см. стр. 46)

Выключение объектов в vCore

Перед началом процесса обновления необходимо выключить все активные ВМ во всех кластерах vCore.

Выключение всех сервисов vCore

Узлы управления

При обновлении управляющей части необходимо строго соблюдать порядок работ на узлах.

Остановите службы системы управления во всех узлах с учётом статуса миграции (статус миграции необходимо проверять в параметре provider.migration.enabled, который указан в файле `/opt/tvc/broker/config/application.properties`):

- вначале control последовательно на всех узлах:

```
systemctl stop tionix-tvc-control
```

- затем broker последовательно на всех узлах. В первую очередь необходимо выключить брокер с включенной миграцией (true), а потом - брокеры с выключённой (false), если таковые имеются:

```
systemctl stop tionix-tvc-broker
```

Также необходимо остановить и удалить ресурс `vcore_balancer` в Pacemaker (ресурс VIP можно не трогать):

```
pcs resource disable vcore_balancer
pcs resource delete vcore_balancer
```

Агент ВУ

Примечание: здесь и далее предполагается, что ОС агента ВУ не будет переустановлена полностью. В случае переустановки ОС на ВУ нужно провести стандартную настройку агента, описанный в документации для 1.2 или 1.3.

Выключите агент ВУ и сервис конфигуратора для tty:

```
systemctl stop tionix-tvc-agent tvc-hyper-configurator
```

Создание копии конфигурации

Узлы управления

На всех узлах управления необходимо сохранить каталоги внутри /opt/tvc каталоги в полном составе и с сохранением прав, например, в каталог /root:

```
mkdir -p /root/vcore_backup
cp -rp /opt/tvc/* /root/vcore_backup
```

При наличии балансера необходимо сохранить конфигурацию этого сервиса во всех узлах:

```
cp -p /etc/sysconfig/tvc-balancer /root/vcore_backup
```

Агент ВУ

На всех узлах ВУ необходимо сохранить каталог /opt/tvc в полном составе и с сохранением прав, например, в каталог /root:

```
cp -rp /opt/tvc/* /root/vcore_backup
```

Удаление пакетов с именем tvc

Узлы управления

Перед установкой новых пакетов необходимо удалить старые пакеты vCore на всех узлах:

```
dnf remove -y tionix-tvc-broker tionix-tvc-control tionix-tvc-balancer \
              tionix-tvc-openjdk libvirt-dbus-tvc
```

Агент ВУ

Перед установкой новых пакетов необходимо удалить старый список пакетов vCore на всех узлах:

```
dnf remove -y tionix-tvc-agent tvc-hyper-configurator libvirt-dbus-tvc
```

Установка новых пакетов

Отмонтирование NFS-каталогов

Перед переносом домашнего каталога необходимо во всех узлах платформы отмонтировать активные NFS-каталоги. Для этого вызовите команду mount с фильтрацией по NFS-каталогам:

```
mount | grep -i nfs
```

Примерный вывод строки активного NFS-каталога:

```
NFS_IP_ADDR:/ on /home/tvc/storages/{...} type nfs4 (parameters)
```

Для отмонтирования используйте полный путь каталога:

```
umount /home/tvc/storages/{...}
```

⚠ При использовании хранилища ВМ на NFS, во избежании блокировок файлов ВМ сервисом sanlock, необходимо убрать сервис агента ВУ из автозапуска и перезагрузить ВУ.

```
systemctl disable tionix-tvc-agent sanlock
```

Изменение имени пользователя и группы

Для сервисов vCore используется пользователь и группа с UID/GID 69, который в старых инсталляциях имел имя tvc. Перед установкой пакетов необходимо поменять имя пользователя и группу на vcore, а также сменить домашний каталог для всех узлов управления и ВУ:

```
usermod -l vcore tvc
groupmod -n vcore tvc
usermod -d /home/vcore -m vcore
```

Добавление репозитория

Узлы управленя

Во всех узлах СУ добавьте новый репозиторий vCore:

```
cat << 'EOF' >> /etc/yum.repos.d/vcore-rpm.repo
[vcore-rpm]
name=vCore Dev Repo
baseurl=https://vcore-public:vcore-public@maven.tionix.ru/artifactory/vcore-rpm/fedora/
linux/$releasever/$basearch
enabled=1
gpgcheck=0
EOF
```

Обновите список репозиториев:

```
dnf update
```

Агент ВУ

Во всех узлах ВУ выполните следующую команду:

```
dnf -y upgrade
```

После окончания обновления перезагрузите узел:

```
reboot
```

Установите пакет dnf-plugin-system-upgrade:

```
dnf -y install dnf-plugin-system-upgrade
```

Скачайте пакеты обновления Fedora 37, выполнив следующую команду:

```
dnf system-upgrade download --releasever=37
```

⚠ В процессе обновления, могут возникнуть проблемы, связанные с неудовлетворенными зависимостями пакетов. Обновление будет прервано, для продолжения, можно запустить команду ещё раз ключём --allowerasing.

После завершения процесса запустите команду перезапуска для принятия изменений:

```
dnf system-upgrade reboot
```

Система перезагрузится и запустится процесс обновления. Может занять длительное время, в течении которого узел будет недоступен.

После загрузки ВУ проверьте версию ОС:

```
cat /etc/fedora-release
Fedora release 37 (Thirty Seven)
```

Удалите текущие репозитории и добавьте новые с пакетами vCore и зеркалом Fedora 37:

```
rm -rf /etc/yum.repos.d/*

cat << 'EOF' >> /etc/yum.repos.d/vcore.repo

[fedora-local]
name=Fedora Local $basearch
baseurl=https://vcore-public:vcore-public@maven.tionix.ru/artifactory/fedora-local/
linux/releases/$releasever/Everything/$basearch/os/
enabled=1
gpgcheck=0

[fedora-modular-local]
name=Fedora Modular Local - $basearch
baseurl=https://vcore-public:vcore-public@maven.tionix.ru/artifactory/fedora-local/
linux/releases/$releasever/Modular/$basearch/os/
enabled=1
gpgcheck=0

[fedora-updates-local]
name=Fedora - $basearch - Updates - Local
baseurl=https://vcore-public:vcore-public@maven.tionix.ru/artifactory/fedora-local/
linux/updates/$releasever/Everything/$basearch/
enabled=1
gpgcheck=0

[fedora-modular-updates-local]
name=Fedora - $basearch - Modular Updates - Local
baseurl=https://vcore-public:vcore-public@maven.tionix.ru/artifactory/fedora-local/
linux/updates/$releasever/Modular/$basearch/
enabled=1
gpgcheck=0

[vcore-engine]
name=vCore Engine
baseurl=https://vcore-public:vcore-public@maven.tionix.ru/artifactory/vcore-rpm/fedora/
linux/$releasever/$basearch
enabled=1
gpgcheck=0

EOF
```

Установка пакетов

Узлы управления

После удаления всех пакетов необходимо установить новые с другим именованием:

```
dnf install -y vcore-broker vcore-control
```

При кластерной установке доустановите пакет с балансером:

```
dnf install vcore-balancer
```

Агент ВУ

Установите новые пакеты с другим именованием:

```
dnf install -y vcore-agent vcore-hyper-configurator vcore-multipath-config
```

Возвращение конфигурации vCore

Узлы управления

После установки пакетов необходимо вернуть конфигурацию сервисов и базу данных состояния платформы на всех узлах.

База данных и статус брокера сообщений

Вначале во всех узлах скопируйте содержимое копии конфигурации, которая для примера была сохранена в каталог /root/tvc, в каталог /opt/vcore:

```
cp -rp /root/vcore_backup/ignite /opt/vcore/ignite
cp -rp /root/vcore_backup/broker/data /opt/vcore/broker
```

Конфигурация сервисов

Из-за изменений в конфигурации необходимо перенести все изменения с сохраненной конфигурации application.properties сервисов broker и control. Копировать конфигурацию как есть нельзя, иначе эти сервисы не запустятся. Лучше всего взять новую конфигурацию и настроить её как при обычной установке с учётом изменений параметров по умолчанию в старой конфигурации.

Все изменения для broker в /root/vcore_backup/broker/config/application.properties нужно применить в файле

```
/opt/vcore/broker/config/application.properties
```

Соответственно для control в /root/vcore_backup/broker/config/application.properties нужно применить в файле

```
/opt/vcore/control/config/application.properties
```

Конфигурацию balancer можно перенести как есть:

```
cp -p /root/vcore_backup/balancer/config/standalone.xml /opt/vcore/balancer/config/
standalone.xml
cp -p /root/vcore_backup/tvc-balancer /etc/sysconfig/vcore-balancer
```

Агент ВУ

Все изменения для agent в /root/vcore_backup/config/application.properties нужно применить в файле

```
/opt/vcore/agent/config/application.properties
```

Копировать конфигурацию как есть нельзя, иначе этот сервис не запустится.

У конфигуратора своего файла параметров нет.

Запуск сервисов

После копирования всей конфигурации необходимо в определенном порядке запустить сервисы vCore.

Узлы управления

Балансер

При кластерной установке в одном из узлов в Pacemaker создайте, определите зависимости, последовательность запуска и активизируйте сервис балансера:

```
pcs resource create vcore_balancer systemd:vcore-balancer op monitor interval="10s"
 timeout="15s"
pcs constraint colocation add vcore_balancer with vcore_vip
pcs constraint order start vcore_vip then start vcore_balancer
pcs resource enable vcore_balancer
```

Проверьте статус кластера:

```
Cluster name: vcorecluster
Cluster Summary:
  * Stack: corosync
  * Current DC: 192.168.122.11 (version 2.1.4-4.fc35-dc6eb4362e) - partition with quorum
  * Last updated: Fri Oct 21 18:53:47 2022
  * Last change:  Fri Oct 21 14:28:46 2022 by root via cibadmin on 192.168.122.33
  * 3 nodes configured
  * 2 resource instances configured

Node List:
  * Online: [ 192.168.122.11 192.168.122.33 192.168.122.153 ]

Full List of Resources:
  * vcore_vip   (ocf::heartbeat:IPaddr2):        Started 192.168.122.33
  * vcore_balancer      (systemd:vcore-balancer):   Started 192.168.122.33
 <------ Ресурс сервиса balancer в статусе Started

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Сервисы control и broker

Запустите службы на всех узлах кластера (статус миграции необходимо проверять в параметре provider.migration.enabled, который указан в файле `/opt/vcore/broker/config/application.properties`):

- Запустите broker **последовательно** на узлах с migration=false:

  ```
  systemctl start vcore-broker
  ```

- На одном из узлов запустите очистку партиций кэшей БД (IP используется для примера):

  ```
  curl -XPUT -H "Content-Type: application/json" http://192.168.122.153:8081/api/
  storage/manage/resetLostPartitions
  ```

- После этого запустите брокера в узле с migration=true:

  ```
  systemctl start vcore-broker
  ```

- После успешного запуска последнего брокера проверьте топологию:

  ```
  curl -X "GET" http://192.168.122.153:8081/api/storage/manage/state | jq .
  ```

- Топология должна выглядеть примерно так (она должна выглядеть одинаковой для всех сервисов брокера):

  ```
  {
    "clusterState": "ACTIVE",
    "baselineAutoAdjustment": false,
    "baselineAutoAdjustmentTimeout": 300000,
    "topologyVersion": 3,
    "nodes": [
      {
        "id": "c997ddb2-af3e-49a3-9dfa-dfebbf89064b",
        "consistentId": "962a7caf-1727-4c94-ba1e-5bf02fd190bc",
        "inTopology": true,
        "offline": false,
  ```

```
      "heapMemMax": 8589934592,
      "heapMemTotal": 8589934592,
      "heapMemUsed": 1581150072,
      "nonHeapMemUsed": 141409672,
      "hostNames": [
        "vcore_ctrl2"
      ]
    },
    {
      "id": "bdb6d879-ec3b-4a55-8a79-03accfd764f6",
      "consistentId": "20a4bd59-4de9-4ea2-9151-12b79fe1f566",
      "inTopology": true,
      "offline": false,
      "heapMemMax": 8589934592,
      "heapMemTotal": 8589934592,
      "heapMemUsed": 1783260184,
      "nonHeapMemUsed": 136189184,
      "hostNames": [
        "vcore_ctrl3"
      ]
    },
    {
      "id": "285f35a4-7f57-4109-b3cc-ab3b6265cd89",
      "consistentId": "f4772434-c51b-49d7-b0ff-8865e4aa6f3e",
      "inTopology": true,
      "offline": false,
      "heapMemMax": 8589934592,
      "heapMemTotal": 8589934592,
      "heapMemUsed": 1668433544,
      "nonHeapMemUsed": 138790096,
      "hostNames": [
        "vcore_ctrl1"
      ]
    }
  ]
}
```


- Запустите control **последовательно** на всех узлах:

```
systemctl start vcore-control
```

Убедитесь, что по порту 8082 (по умолчанию) доступен веб-интерфейс control.

Агент ВУ

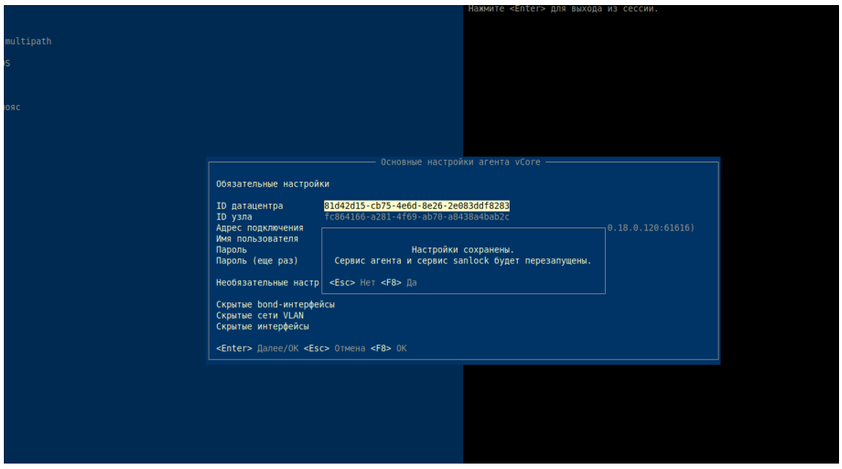
Перед запуском поменяйте конфигурацию /root/.bashrc заменив путь /usr/bin/tvc-hyper-configurator на /usr/bin/vcore-hyper-configurator

```
sed -i 's/tvc/vcore/g' /root/.bashrc
```

Запустите конфигуратор:

```
/usr/bin/vcore-hyper-configurator
```

Авторизуйтесь, в окне основных настроек нажмите F8, агент и сервис sanlock будут запущены:



В списке узлов агенты ВУ должны вернуть себе статусы "Активен" или "Обслуживание".

⚠ При наличии ВМ, созданных из Шаблона и Типом виртуальной машины Зависимая ВМ, для сохранения обратной совместимости, необходимо в разделе /home/ создать симлинк tvc на /home/vcore и поменять его владельца на vcore.

```
cd /home
ln -s /home/vcore tvc
chown -h vcore:vcore tvc
```

Удаление Базис.vCore v.1.3

Для удаления платформы Базис vCore необходимо удалить пакеты сервисов на всех управляющих и вычислительных узлах. Перед удалением платформы рекомендуется сохранить конфигурацию сервисов, так как она будет удалена вместе с пакетами сервисов. Расположение конфигурационных файлов можете получить в разделе - Установка Базис vCore[7].

Управляющая часть

Остановите службы системы управления:

```
systemctl stop vcore-control
systemctl stop vcore-broker
```

Удалите пакеты системы управления:

```
dnf remove vcore-*
```

Вычислительная часть

Остановите службу агента:

```
systemctl stop vcore-agent
```

Удалите агент:

```
dnf remove vcore-*
```

7 https://conf.tionix.ru/pages/viewpage.action?pageId=290095209&src=contextnavpagetreemode

Приложения к руководству по установке Базис.vCore v.1.3

Приложение 1. Список портов сервисов Базис.vCore v.1.3

Сервис	Протокол	Порт
vcore-broker	tcp	8081
	tcp	47400
	tcp	10800
	tcp	47500-47510
	tcp	47100
	tcp	61616
	tcp	61617
	udp	9876
	udp	23365
	udp	47400
	udp	30000-60000
vcore-control	tcp	8082
	udp	23364
vcore-balancer	tcp	8080
	tcp	8081
	tcp	8090
	tcp	8091
	tcp	9990
vcore-agent	tcp	8083
	tcp	16509
	udp	9876

Приложение 2. Рекомендации по настройке систем хранения для хранилищ Базис.vCore v.1.3

- Введение (см. стр. 52)
- NFS (см. стр. 52)
- iSCSI (см. стр. 53)
- Fibre Channel (см. стр. 54)

Введение

Данная статья содержит информацию с примерами настройки различных типов систем хранения для их добавления в качестве хранилища для vCore.

NFS

В качестве примера настройки приведём конфигурацию NFS-сервера в ОС Linux.

> **Важно**
>
> Не запускайте NFS-сервер на управляющих узлах vCore.
> NFS-сервер должен поддерживать параметры идентификаторов пользователей и группы (anonuid, anongid), так как vCore не поддерживает запись на каталог NFS от имени root.

Установите необходимые пакеты:

```
dnf install nfs-server nfs-utils
```

Создайте директорию для ISO-хранилища:

```
mkdir /srv/iso
```

Создайте директорию для хранилища данных:

```
mkdir /srv/volumes
```

Дайте права доступа к хранилищам:

```
chown vcore:vcore /srv/iso /srv/volumes
```

Настройте параметры хранилищ в файле - `/etc/exports`:

```
/srv/iso IP_ADDR/24(rw,sync,anonuid=69,anongid=69,all_squash)
/srv/volumes IP_ADDR/24(rw,sync,anonuid=69,anongid=69,all_squash)
```

Где:

- IP_ADDR - подсеть сети хранения vCore;
- rw,sync,anonuid=69,anongid=69,all_squash - параметры экспорта NFS-каталога.

Запустите NFS-сервисы:

```
systemctl restart nfs-utils
systemctl restart nfs-server
```

Добавьте сервисы в автозагрузку:

```
systemctl enable nfs-utils nfs-server
```

Проверьте, что каталоги NFS добавлены:

```
exportfs
```

Для добавления хранилища в vCore воспользуйтесь инструкцией - Настройка хранилища с сетевой файловой системой (NFS)[8].

iSCSI

Установите пакет `targetcli`:

```
dnf -y install targetcli
```

Посмотрите расположение дискового устройства, который необходимо подключить к платформе:

```
fdisk -l
```

Пример ответа:

```
Disk /dev/vdb: 500 GiB, 536870912000 bytes,
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Откройте targetcli:

```
targetcli
```

Перейдите в директорию - `/backstores/block`:

```
cd /backstores/block
```

Создайте привязку луна `lun1` к физ устройству - `/dev/vdb`:

```
/backstores/block> create lun1 /dev/vdb
```

Перейдите в директорию `/iscsi`:

```
cd /iscsi
```

Создайте iSCSI-таргет:

```
create
```

Пример ответа:

```
Created target iqn.2003-01.org.linux-iscsi.vcore-hv-8add8fd1.x8664:sn.95b3c20c7158.
Created TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260.
```

Перейдите в директорию - `/iscsi/iqn.2003-01.org.linux-iscsi.vcore-hv-8add8fd1.x8664:sn.95b3c20c7158/tpg1/luns`:

```
cd /iscsi/iqn.2003-01.org.linux-iscsi.vcore-hv-8add8fd1.x8664:sn.95b3c20c7158/tpg1/luns
```

Создайте привязку таргета `iqn.2003-01.org.linux-iscsi.vcore-hv-8add8fd1.x8664:sn.95b3c20c7158` к `lun1`:

```
create /backstores/block/lun1
```

8 https://conf.tionix.ru/pages/viewpage.action?pageId=375882161#id-УправлениехранилищамиБазис.vCorev.1.3-Настройкахранилищассетевойфайловойсистемой(NFS)

Перейдите в директорию - `/iscsi/iqn.2003-01.org.linux-iscsi.vcore-hv-8add8fd1.x8664:sn.95b3c20c7158/tpg1/acls`:

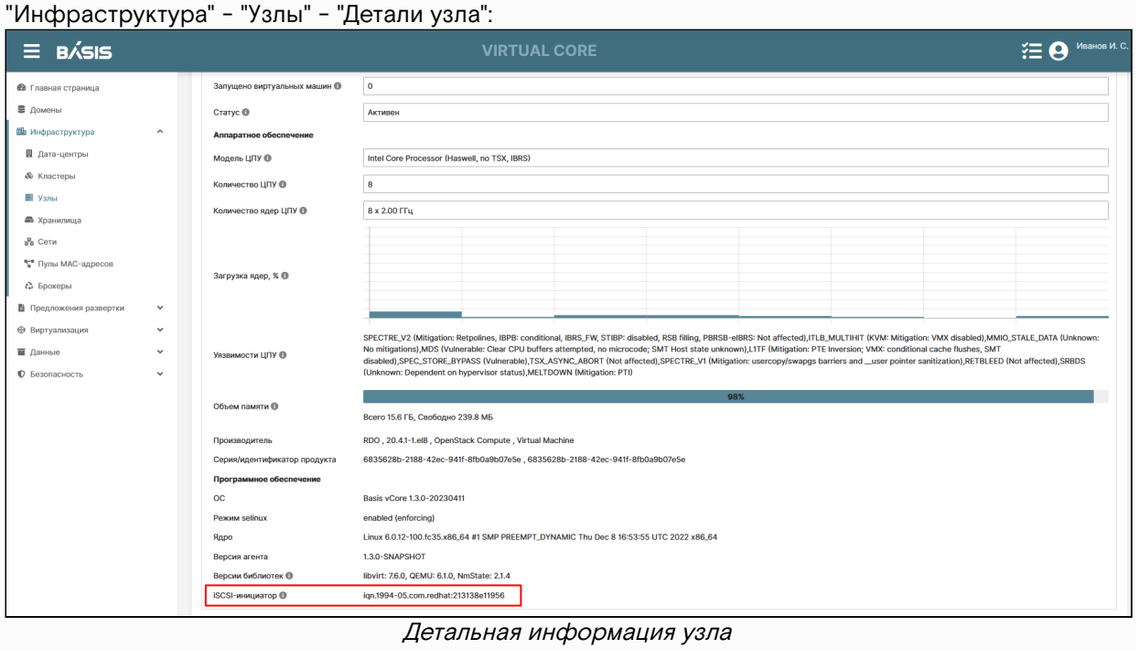
```
cd /iscsi/iqn.2003-01.org.linux-iscsi.vcore-hv-8add8fd1.x8664:sn.95b3c20c7158/tpg1/acls
```

Создайте привязку iSCSI-инициатора:

```
create iqn.1994-05.com.redhat:213138e11956
```

Где: `iqn.1994-05.com.redhat:213138e11956` - iSCSI-инициатор узла, через который будет происходить управление хранилищем.

iSCSI-инициатор узла можно посмотреть на панели управления платформой vCore, во вкладке - "Инфраструктура" - "Узлы" - "Детали узла":



Детальная информация узла

Для сохранения изменений в `targetcli` выполните:

```
exit
```

Для добавления хранилища в vCore воспользуйтесь инструкцией - Настройка iSCSI-хранилища[9].

Fibre Channel

Особых настроек для работы Fibre Channel с vCore не требуется. Для корректной работы Fibre Channel необходимо наличие HBA-контроллеров на всех вычислительных узлах vCore и настроенный конфигурационный файл для multipath.

Процесс настройки системы хранения зависит от конкретного оборудования. За подробной инструкцией обратитесь к производителю, чьё оборудование используется для хранилищ vCore.

9 https://conf.tionix.ru/pages/viewpage.action?pageId=375882161#id-УправлениехранилищамиБазис.vCorev.1.3-НастройкаiSCSI-хранилища