Почтовая система RuPost

Руководство по установке и конфигурированию

RU.47022019.10001-01 91 01

Версия 4.2.0 – Май 2026

Официальный веб-сайт: https://www.rupost.ru/

Оглавление

1. Обзор системы ... 8
1.1. Функциональные возможности RuPost ... 8
1.2. Архитектура и компоненты сервера RuPost ... 9
1.3. Отказоустойчивый кластер RuPost ... 12
1.3.1. Улучшение масштабируемости при развертывании узлов кластера на виртуальных машинах ... 15
1.3.2. Ребалансировка распределения подключений IMAP по узлам кластера ... 18
2. Подготовка к установке RuPost ... 19
2.1. Системные требования ... 19
2.2. Операционная система ... 19
2.3. Синхронизация времени ... 20
2.4. Службы каталогов LDAP ... 20
2.4.1. Служебная учётная запись ... 21
2.4.2. Интеграция с ALD Pro ... 21
2.4.3. Подготовка FreeIPA ... 29
2.4.4. Подготовка Microsoft Active Directory ... 29
2.4.5. Поддержка Samba DC ... 31
2.4.6. Подготовка Avanpost DS ... 32
2.5. Система управления базами данных ... 33
2.5.1. Настройка Tantor BE/SE ... 34
2.5.2. Общие настройки для PostgreSQL, Tantor BE/SE, кластера на основе patroni ... 34
2.6. Служба кэширования объектов в оперативной памяти rupost-cs (Memcached) ... 41
2.7. Пространства хранения, группы ящиков и хранилища ... 43
2.8. Подключение сетевых каталогов файловой системы NFSv4 ... 45
2.8.1. Рекомендации по настройке сетевых файловых хранилищ ... 47
2.8.2. Рекомендации по настройке сетевого файлового хранилища на примере NFS сервера в составе ОС Astra Linux ... 55
2.8.3. Рекомендации по настройке количества inode для хранилищ почтовых ящиков ... 56
2.9. Резервный NFS для очередей Postfix ... 57
2.10. Настройки DNS ... 58
3. Установка RuPost ... 61
3.1. Установка системы с помощью мастера установки ... 62
3.2. Командный интерфейс мастера установки (CLI) ... 66

4. Обновление системы ... 68
4.1. Непрерывное обновление кластера ... 69
5. Подготовка системы к реальной эксплуатации (меры информационного контроля) ... 73
5.1. Использование действительных корпоративных сертификатов ... 73
6. Действия после установки и настройки системы ... 75
7. Удаление RuPost из операционной системы ... 76
8. Основные пути и файлы системы ... 77
9. Средства диагностики ... 79
9.1. Единый сводный журнал (лог) для всех почтовых компонентов – команда CLI logs ... 79
9.2. Поддержка сбора и экспорта логов – команда CLI report ... 80
9.3. Поддержка SOSReport ... 81
10. Приложение 1. Расчёт системных требований в зависимости от планируемой нагрузки ... 82
10.1. Общие замечания к расчёту ожидаемых системных требований ... 82
10.2. Оперативная память ... 82
10.3. Процессор ... 83
10.4. Дисковая память ... 83
10.5. Подключения к базе данных PostgreSQL ... 83
10.6. Пример расчёта минимальных системных требований ... 83
11. Приложение 2. Функциональное взаимодействие RuPost с подключенными доменами LDAP ... 85
11.1. Права доступа к атрибутам у служебной учётной записи RuPost ... 85
11.1.1. FreeIPA ... 85
11.1.2. ALD Pro ... 85
11.1.3. Active Directory ... 86
11.2. Функциональное использование объектных классов и атрибутов LDAP ... 87
11.2.1. Классы ... 87
11.2.2. Ключевые атрибуты ... 87
11.3. Атрибуты для глобальной адресной книги ... 88
11.3.1. FreeIPA ... 88
11.3.2. ALD Pro ... 89
11.3.3. Active Directory ... 89
11.3.4. Samba DC ... 90
11.3.5. Avanpost DS ... 90
12. Приложение 3. Сетевые настройки (порты) ... 92

Внимание!

Перед обновлением версии RuPost выполните **резервное копирование** узлов кластера и баз данных.

Внимание!

Перед началом установки RuPost должны быть подключены и доступны «**base**» и «**extended**» репозитории AstraLinux.
При возникновении вопросов, связанных с обновлением операционной системы, обращайтесь в техподдержку ГК Астра.

Внимание!

Перед обновлением версии RuPost обратите внимание на выбор варианта обновления:

- **обычное** – все узлы кластера обновляются одновременно, но требуется предварительный вывод из эксплуатации всех экземпляров RuPost;

- **непрерывное** – обновление без прерывания обслуживания пользователей с последовательным обновлением узлов кластера.

Рекомендуется

Проводить непрерывное обновление в период, когда нагрузка на почтовую систему снижается.

Внимание!

После завершения **непрерывного** обновления возможно неравномерное распределение пользователей по узлам кластера.

При необходимости, для перераспределения пользователей на менее загруженные узлы, на всех узлах с большим количеством пользователей выполните команду:

```
rupost kick-local-users
```

Внимание!

При проведении обновления в обычном режиме (с предварительным выводом всех узлов из эксплуатации) – проверьте, завершилась ли успешно установка новой версии на всех узлах кластера перед вводом узлов в эксплуатацию. Если на каком-то из узлов новая версия не установилась, то необходимо установить RuPost на этот узел вручную.

Внимание!

Если у вас в организации используются почтовые клиенты Desktop X и Workspad X рекомендуем, после завершения обновления RuPost, обновить сервер Workspad на актуальную версию.
Если устанавливается WorksPad сервер версии 7.0 и выше, то нужно в настройках RuPost указывать адрес WorksPad сервера в новом формате:
https://<WorksPadGateway>/rupostapi

Рекомендуется

Так как в версии 4.2.0 была изменена структура Архивов, то при обновлении с более ранних версий рекомендуется, после обновления всех узлов кластера и разворачивания обновлённой конфигурации, выполнить скрытую команду:

```
rupost mailbox update-archive-emails-folders
```

Выполнение команды может занять продолжительное время. Её результатом станет перестраивание структуры каталогов архивной почты, которая переносится в архивное хранилище автоматически в соответствии с политикой архивации.

Внимание!

Начиная с версии 3.2.0 в процедуру авторизации пользователей добавлена проверка корректности заполнения атрибута proxyAddresses. Просьба проверить, что у всех пользователей в атрибуте proxyAddresses корректно заполнена информация о первичном почтовом адресе - в формате "SMTP: e-mail@domain.ru".
При необходимости, восстановить корректные значения атрибутов LDAP для отдельного почтового ящика можно из окна свойств почтового ящика, нажав на кнопку «Сохранить».

Внимание!

После обновления с предыдущей версии необходимо повторно развернуть активную или выбрать новую конфигурацию на основании обновленных шаблонов конфигураций, устанавливаемых при обновлении системы.

Внимание!

После завершения установки версии 4.1.1 и выше, для обеспечения корректного ведения лога почтовых компонентов, необходимо выполнить команду:

```
sudo /usr/sbin/syslog-ng-ctl reload
```

Рекомендуется

Обновить операционную систему на узлах RuPost и серверах NFS до версии AstraLinux 1.8.5.

Внимание!

Если вы используете **custom шаблон** конфигурации с секцией configuration:haproxy, то перед развертыванием конфигурации (в связи с обновлением механизма кластерного healthcheck), необходимо обновить используемые custom шаблоны следующим образом:
Если вы не вносили изменения в секцию haproxy в своих шаблонах, то в ваш шаблон необходимо скопировать обновленную секцию haproxy из /var/lib/rupost/templates/basic_astra17.yml.
Если вы вносили изменения в секцию haproxy, то ваши изменения необходимо повторить, взяв за основу обновленную секцию haproxy из /var/lib/rupost/templates/basic_astra17.yml.

Обновлённый шаблон необходимо добавить в библиотеку шаблонов и развернуть конфигурацию используя этот шаблон.

1. Обзор системы

RuPost – почтовая система, предназначенная для предприятий любого масштаба – от небольших организаций до корпораций. RuPost устанавливается в корпоративной сети предприятия и работает на платформе Astra Linux.

1.1. Функциональные возможности RuPost

RuPost включает следующую функциональность:

- Панель управления почтовой системой, доступная через современные web-браузеры
- Командный интерфейс управления (CLI)
- Электронная почта (протоколы SMTP, IMAP, POP3)
- Календари (протокол CalDav)
- Задачи (протокол CalDav)
- Контакты (протокол CardDav)
- Корпоративная адресная книга (на базе LDAP)
- Списки рассылки (статические и динамические)
- Ресурсы календаря
- Почтовые правила
- Глобальные правила фильтрации почты
- Полномочия и разрешения
- Аудит действий администраторов
- Выгрузка системной информации и логов почтовых компонент через CLI
- Поддержка SOSReport
- Мониторинг изменений в конфигурационных файлах почтовых компонентов с возможностью автоматического восстановления
- Интеграция с корпоративными службами каталогов LDAP – Active Directory, FreeIPA, ALDPro
- Интеграция со средствами информационного контроля по протоколу Milter (антивирусная/антиспам/антималваре защита, DLP) – включая интеграцию “из коробки” с Kaspersky Security (KLMS, KSMG) и Dr.Web
- Использование СУБД PostgreSQL/Postgres Pro/Tantor
- Встроенный Web-клиент
- Поддержка настольных почтовых клиентских приложений
    - Кроссплатформенный настольный почтовый клиент RuPost Desktop X
    - Microsoft Outlook со специальным модулем расширения RuPost с поддержкой календарей, контактов, задач и адресных книг RuPost по протоколам CalDav и CardDav
    - Evolution и его расширенная версия для Astra Linux
    - Thunderbird и настольные почтовые клиенты на его основе (например, МойОфис, Р7)
- Поддержка контролируемого мобильного доступа с использованием WorksPad (отдельный продукт, интегрируемый с RuPost, включая специальную поддержку пуш-уведомлений)
- Средства миграции почтовых ящиков, календарей и контактов с Microsoft Exchange, с возможностью сосуществования почтовых систем RuPost и Exchange в одном почтовом домене на период миграции.

Система RuPost развертывается в корпоративной сети предприятия (on-prem) или в частном облаке и управляет почтовыми ящиками только для пользователей зарегистрированных в системе доменов LDAP. Регистрация доменов LDAP производится в Панели управления RuPost. Администраторами системы могут выступать только пользователи LDAP либо локальный администратор ОС.

RuPost имеет следующие редакции лицензий:

- Стандартная – Standard
- Корпоративная – Enterprise

В версии 2.5.4 добавлена поддержка лицензий следующих редакций:

- Standard Education
- Enterprise Education
- Enterprise Education Upgrade
- Standard Student
- Enterprise Student
- Enterprise Student Upgrade

Корпоративная редакция (Enterprise) отличается расширенными функциональными возможностями и может устанавливаться в отказоустойчивой (кластерной) конфигурации.

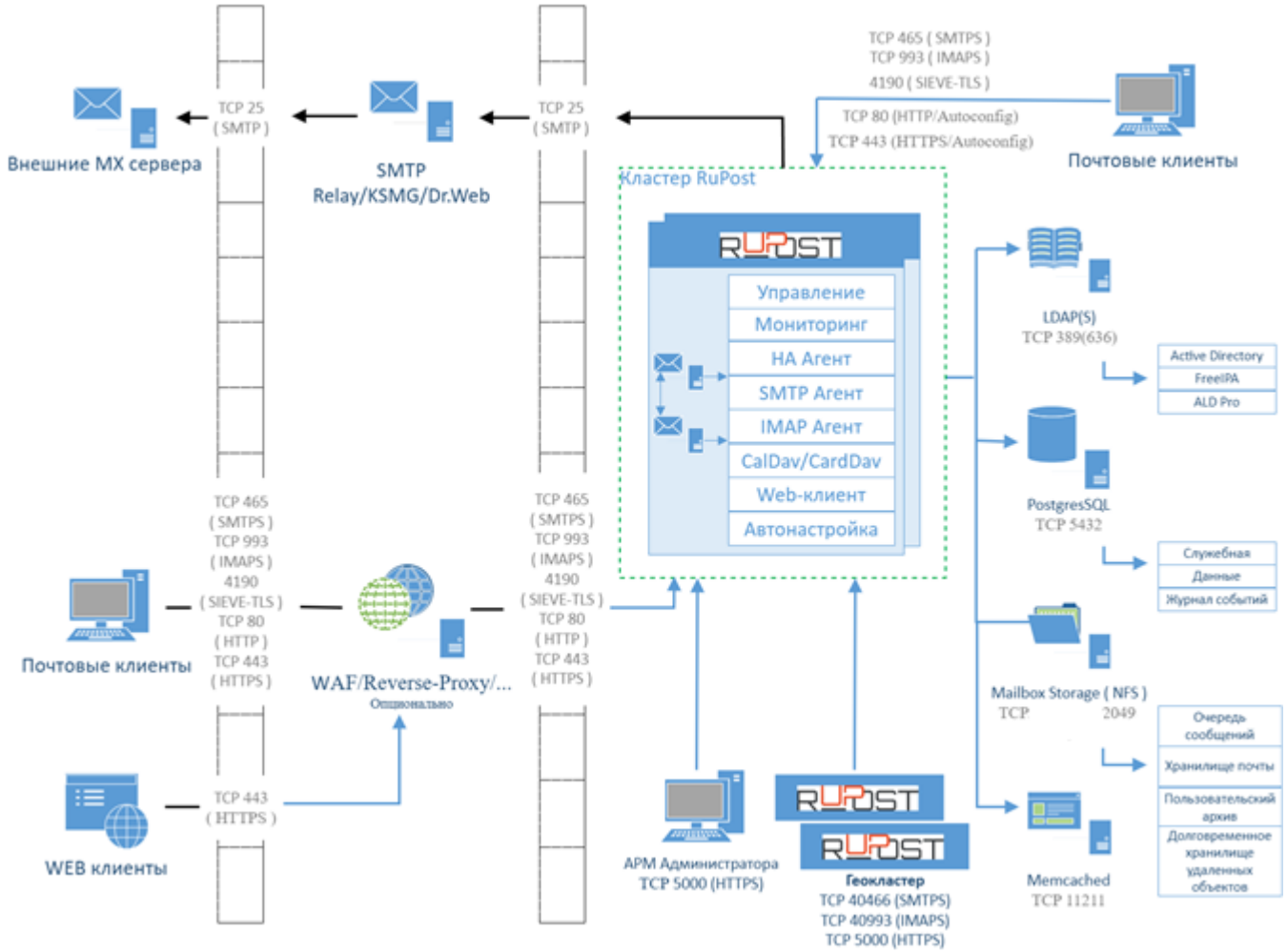
1.2. Архитектура и компоненты сервера RuPost

Сервер RuPost включает в себя систему управления, средства мониторинга, интегрированный набор почтовых компонентов для работы по протоколам SMTP/IMAP/POP3/CalDAV/CardDAV, вспомогательные сервисы, веб-клиент и средства автонастройки клиентских приложений, предназначенные для организации электронной почты корпоративного класса.

Сервер RuPost может функционировать как на одном узле, так и в кластере из множества узлов.

Кластер RuPost предназначен для обеспечения высокой доступности почтовой системы. Кластер функционирует в режиме Active-Active, где все экземпляры системы равнозначны и динамически перераспределяют нагрузку между собой. Каждый узел системы в кластере обладает всеми функциями управления и даже при полном выводе из эксплуатации узла или выходе из строя любого из его компонент продолжит функционировать пока в системе есть хоть один функционирующий узел. Сбои отдельных компонентов обнаруживаются автоматически и соответствующие узлы также автоматически выводятся из эксплуатации, при этом кластер продолжает функционировать.

Конфигурационные параметры системы хранятся в единой БД и совместно используются всеми экземплярами – узлами системы.



Внутренние компоненты сервера RuPost:

- **Система управления** RuPost – ядро системы, обеспечивающее функции управления, мониторинга и автонастройки, инструментарий командной строки и визуальную Панель управления системой (АРМ Администратора):
  - управление и применения типовых шаблонов конфигураций к почтовым компонентам;
  - настройка, конфигурирование, мониторинг, диагностика и управление поведением системы и ее почтовых компонентов через специально разработанные адаптеры почтовых компонентов;
  - подключение пользователей и управление почтовыми адресами и ящиками;
  - управление квотами и другими параметрами *пользовательских ящиков;*
  - *управление администраторами системы RuPost;*
  - *управление обслуживаемыми почтовыми доменами;*
  - *подключение к службам каталогов Active Directory и другими LDAP;*
  - автоматическое формирование и обновление корпоративной адресной книги (Global Address List, GAL) на базе информации из подключенных служб каталогов;
  - кластеризация узлов для обеспечения отказоустойчивости и высокой доступности системы;
  - проверка работоспособности и целостности системы со встроенным мониторингом и самодиагностикой узлов системы, и её компонентов на каждом узле;
  - журналирование операций;

  - графическая **Панель управления** RuPost, доступная из браузера;
  - командный интерфейс управления (CLI).

- **HA Агент (High Availability Agent)** – **Rupost-lb** (HAProxy), агент обеспечения высокой доступности системы, работающий по протоколам TCP, HTTP(S). Выполняет следующие функции:
  - дублирующее отслеживание состояния Системой управления и других HA Агентов системы;
  - в случае сбоя работы почтовых сервисов на узле - перенаправление сетевых запросов от пользователей на другие узлы кластера;
  - проксирование и терминирование соединений к почтовым компонентам по протоколам IMAP и SMTP;
  - контролируемый доступ клиентских приложений к почтовым ящикам;
  - IMAPS, SMTPS, POP3S и контролируемый доступ к web-клиенту с использованием SSL/TLS.

- **SMTP Агент – RuPost-mta** (на основе Postfix). Компонент пересылки писем (Mail Transfer Agent, MTA), работающий по протоколам TCP, SMTP, LMTP, STARTTLS, TLS, SASL, LDAP, Milter. Номера занимаемых портов зависят от типа конфигурации. Основными задачами данного сервиса являются:
  - получение писем от сторонних почтовых серверов;
  - отправка писем пользователей сторонним почтовым серверам;
  - передача полученных писем компоненту обработки писем MDA Dovecot по протоколу LMTP для дальнейшего сохранения в пользовательских почтовых ящиках и/или отправки конечным адресатам;
  - получение пользовательских писем от Mail User Agent (MUA) для последующей пересылки сторонним почтовым серверам или пользователям своего домена;
  - интеграция со средствами контроля (например, Kaspersky Security) для фильтрации входящей и исходящей почты и соединений по протоколу Milter.

- **IMAP Агент – Rupost-mda** (на основе Dovecot)**.** Компонент обработки писем (Mail Delivery Agent, MDA), работающий по протоколам TCP, IMAP, POP3, LMTP, STARTTLS, TLS, SASL, LDAP. Номера занимаемых портов зависят от типа конфигурации. Компонент выполняет следующие функции:
  - предоставление доступа пользователям к личным почтовым ящикам посредством клиентских приложений;
  - осуществление квотирования ресурсов пользовательских ящиков;
  - хранение и управление письмами;
  - обработка пользовательских и глобальных сценариев, написанных на языке Sieve;
  - предоставление средств удалённого изменения пользовательских Sieve сценариев.

  IMAP – основной протокол работы с почтовыми ящиками. Поддержка POP3 является опциональной в дополнение к IMAP для возможности интеграции с унаследованными системами.

  В версии 2.6.0 в конфигурацию MDA по умолчанию добавлено минутное кэширование хэшей данных почтовых ящиков, аутентифицированных в LDAP.

- **CalDAV/CardDAV компонент для календарей и контактов – Rupost-mua** (на основе SOGo). Отвечает за хранение и удалённый доступ к корпоративным и пользовательским календарям, задачам, контактам и корпоративной адресной книге. Работает по протоколам по протоколам CalDAV и CardDAV.

- **Web-клиент корпоративной почты – Rupost-mua** (на основе SOGo). Доступен во всех актуальных версиях современных браузеров.
- **Web-сервер – Rupost-mp** (Nginx). Обеспечивает работу web-клиента почты.
- **Компонент кеширования в оперативной памяти – Rupost-cs** (Memcached). Работает по протоколу TCP. Выполняет функцию кэширования и синхронизации части пользовательских данных, для увеличения быстродействия доступа к календарям, контактам и web-клиенту.
- **Компонент Rupost-dcp** (на основе PgPool), обеспечивающий управление пулом соединений к базе данных и кеширование результатов запросов SoGo при работе с глобальной адресной книгой.

Управление всеми компонентами системы осуществляется через специализированные адаптеры, обеспечивающие интегрированность и целостность конфигураций RuPost.

Концепция управления RuPost строится на использовании **шаблонов конфигураций**, разрабатываемых на основе заранее созданных и проверенных типовых конфигураций интегрированных компонентов. Шаблоны конфигураций описываются на языке YAML, в котором отражаются основные параметры компонентов RuPost. RuPost предоставляет **библиотеку шаблонов конфигураций**, на основе которых развертываются конкретные конфигурации.

Шаблоны конфигураций бывают двух типов:

- **Встроенные (builtin)** – поставляются в составе RuPost
- **Специализированные (custom)** – разрабатываются в рамках проектов внедрения RuPost для учета особенностей требований конкретной организации и ее корпоративного ИТ ландшафта. Такие шаблоны поддерживаются только в старших редакциях продукта RuPost и не поддерживаются в RuPost Standard. Специализированные шаблоны конфигураций могут быть загружены в библиотеку шаблонов с использованием соответствующих инструментов RuPost. Структура шаблонов конфигураций описана в отдельном “*Руководстве по шаблонам конфигураций*” RuPost.

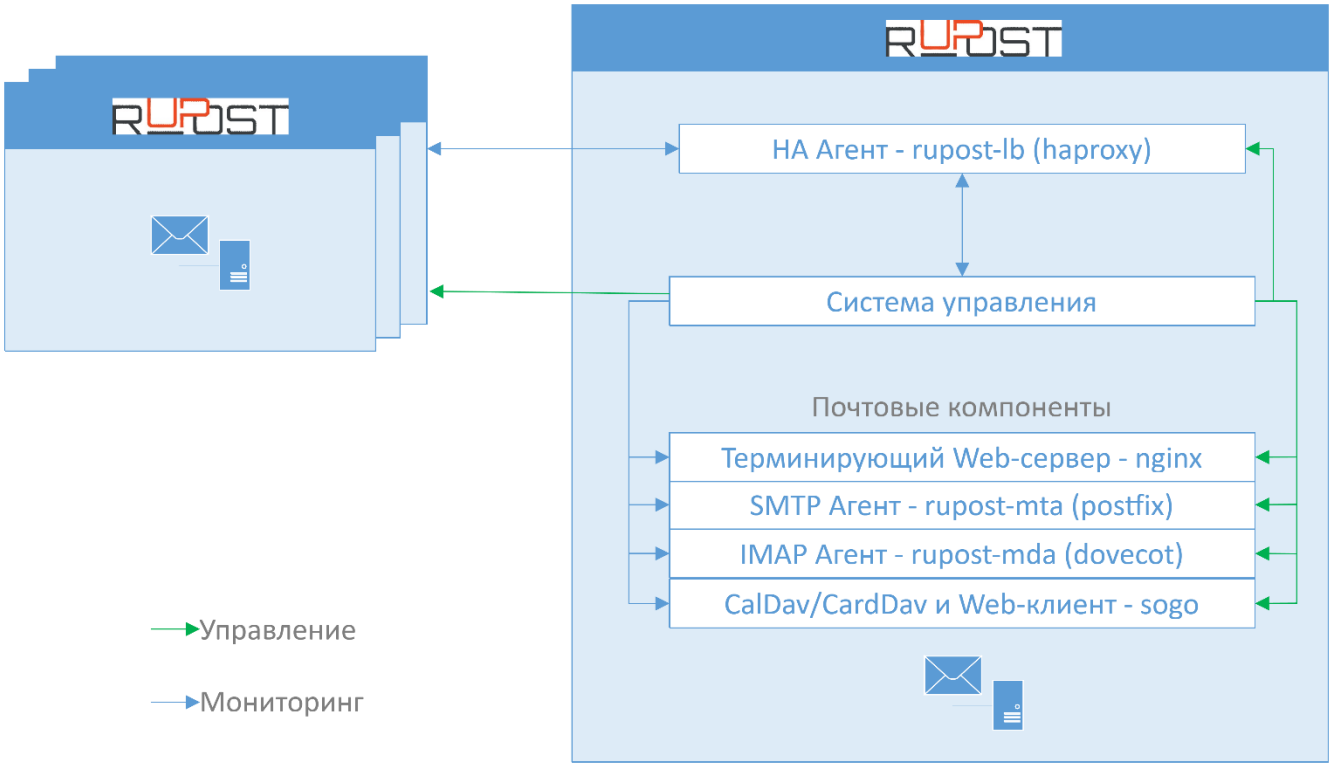
При **развертывании конфигурации** на базе выбранного шаблона система управления RuPost генерирует все необходимые конфигурационные файлы для компонентов системы.

1.3. Отказоустойчивый кластер RuPost

Кластер системы включает **узлы** кластера, на каждом из которых установлен **экземпляр** системы. Все экземпляры системы равнозначны и включают:

- Систему управления с входящей в нее Панелью управления
- HA Агент, обеспечивающий коммуникации с другими HA Агентами и перенаправление полезного трафика на почтовые компоненты
- Почтовые компоненты, к которым относятся:
    - терминирующий Web-сервер – Rupost-mp (nginx)
    - SMTP Агент – Rupost-mta (postfix)
    - IMAP Агент – Rupost-mda (dovecot)
    - CalDav/CardDav сервер с входящим в него почтовым Web-клиентом системы – Rupost-mua (sogo)

Узел доступен для управления и мониторинга, когда на нем функционируют как минимум Система управления и HA Агент.



Экземпляр системы **введен в эксплуатацию**, то есть является функционирующим элементов кластера, при двух условиях:

- для всех компонентов успешно развернута активная конфигурация системы
- все компоненты запущены и функционирую штатно

Все почтовые компоненты (вместе с терминирующим их Web-сервером) работают как единое целое - в режиме синхронизации. То есть при остановке любого из этих компонентов останавливаются они все, причем вне зависимости от того останавливаются они явно администратором или останавливается какой-либо компонент при наличии тех или иных сбоев. Такое поведение почтовых компонентов обеспечивается Системой управления.

Отказоустойчивая архитектура RuPost позволяет обеспечить постепенное масштабирование системы от одного узла до необходимого числа узлов кластера с автоматическим применением одной и той же активной конфигурации RuPost без необходимости индивидуального изменения конфигураций узлов. Число узлов кластера архитектурно неограниченно.

При планировании развертывания системы администратор должен руководствоваться *следующей схемой взаимодействия компонентов и узлов кластера*, обеспечивая открытие необходимых сетевых портов:

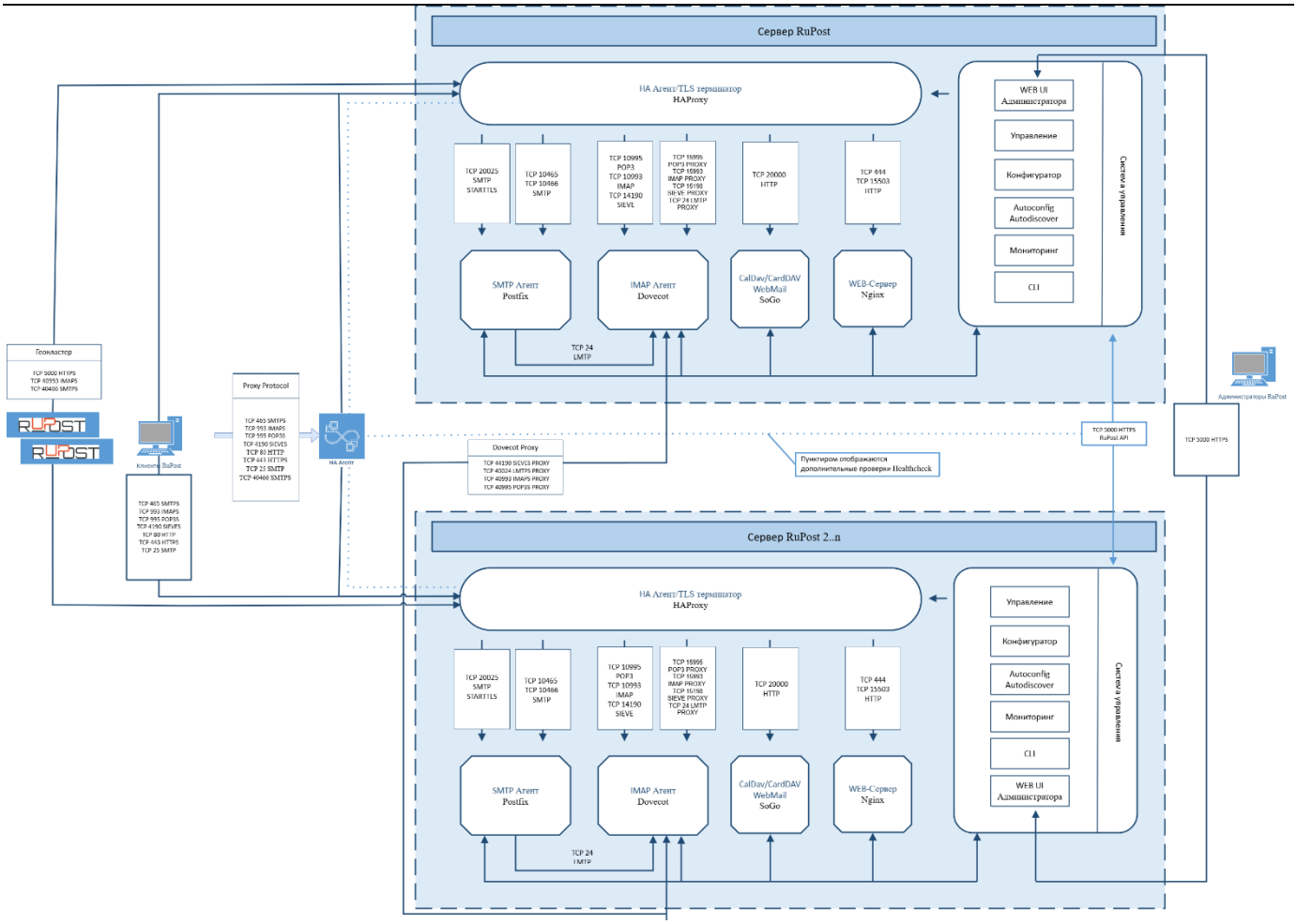


Схема взаимодействия компонентов и узлов кластера

Если клиент при подключении попал на узел, чей экземпляр выведен из эксплуатации – клиент будет перенаправлен на другой узел. В случае идентификации отказа почтовых компонентов любого узла кластера, ассоциированные с ним почтовые очереди автоматически эвакуируются на другой доступный штатно функционирующий узел кластера. Такая высокая доступность (высокий уровень отказоустойчивости) кластера RuPost достигается за счет постоянных проверок работоспособности почтовых сервисов не только средствами сервиса RuPost, но и средствами HAProxy.

Балансировка нагрузки между узлами системы в кластере RuPost может осуществляться с использованием следующих методов:

- Round Robin DNS с использованием A записи, указывающей на набор IP адресов узлов кластера
- Direct Routing
- TPROXY
- Сетевой балансировки с использованием PROXY protocol (v2)

Внимание!
Балансировка TCP трафика на портах 25, 80, 443, 465, 993, 995, 4190 должна выполняться на транспортном уровне алгоритмом Round Robin.

Внимание!
При развертывании кластера рекомендуется вначале развернуть один узел системы, настроить необходимые параметры системы, развернуть на этом узле требуемую конфигурацию и убедиться в работоспособности настроек инфраструктуры и системы, а также доступность системы из клиентских почтовых приложений.
Только после успешного развертывания одного узла стоит переходить к развертыванию и включению в кластер других узлов системы, к которым автоматически будет применяться активная конфигурация по мере их включения в кластер.

Такой подход позволяет сразу убедиться в корректной организации и настройке инфраструктурного ландшафта, необходимого для работы RuPost.

Внимание!
Кластер RuPost является кластером с равнозначными узлами (Active-Active). Разнесение узлов кластера между разными сегментами сети должно производится только при условии обеспечения одинаковой скорости доступа к LDAP, СУБД и системе хранения.

Внимание!
Для обеспечения бесперебойной работы, при установке RuPost будет произведено удаление программы logcheck. Данное действие вызвано тем, что, при отправке почтовых сообщений, logcheck использует нештатный механизм доступа к почтовой системе, что приводит к сбоям в работе RuPost.

1.3.1. Улучшение масштабируемости при развертывании узлов кластера на виртуальных машинах

В версии 4.1.0 добавлена возможность оперативного добавления узлов в кластер RuPost с использованием снимка (“snapshot”) виртуальной машины узла.

Если узлы кластера RuPost развернуты на виртуальных машинах, то, для использования возможности оперативного увеличения количества узлов кластера, администратор имеет возможность сохранить снимок виртуальной машины одного из узлов кластера. Затем, когда возникнет необходимость увеличить вычислительную мощность кластера, достаточно будет добавить новую виртуальную машину на основе ранее сохраненного снимка – узел будет добавлен в кластер автоматически.

Внимание!
Все дальнейшие действия необходимо выполнять из консоли узла источника и клонированного узла. Удаленный доступ к узлу будет недоступен из-за отключения сетевого интерфейса для того, чтобы избежать ситуации с конфликтом IP адресов при клонировании.

Внимание!
Перед созданием снимка - вывести экземпляр RuPost на узле из эксплуатации и отключить виртуальную машину.

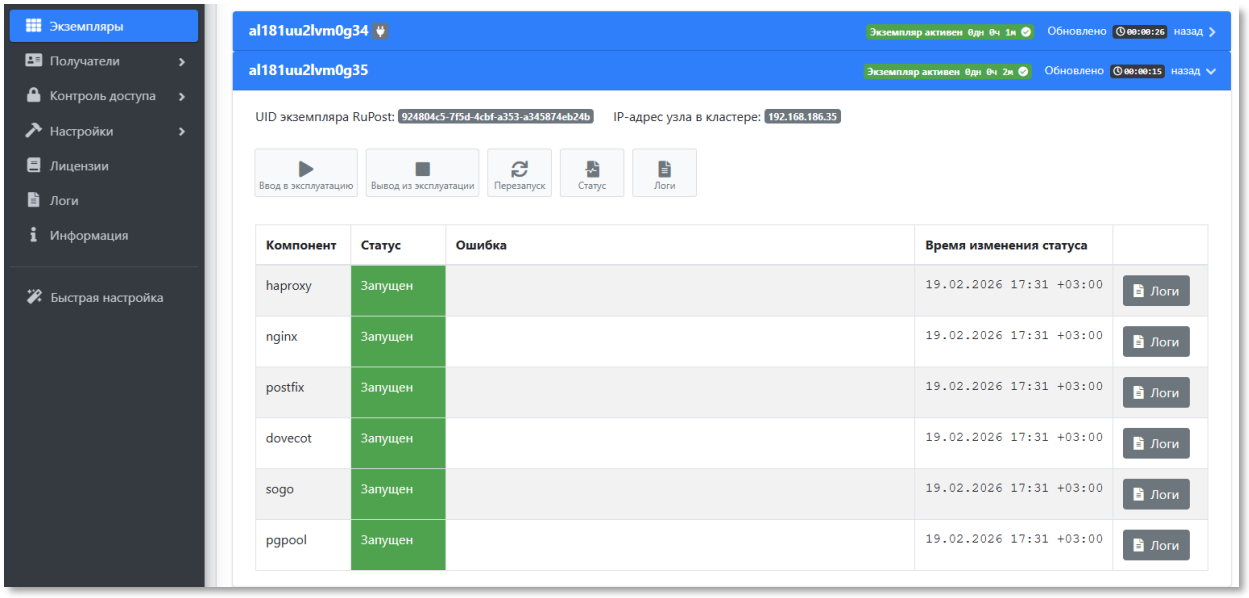
Внимание!
Перед включением виртуальной машины (сделанной на основе снимка) рекомендуется убедиться в отсутствии конфликта IP-адресов с другими узлами кластера RuPost и, при необходимости, назначить этой виртуальной машине новый IP-адрес.

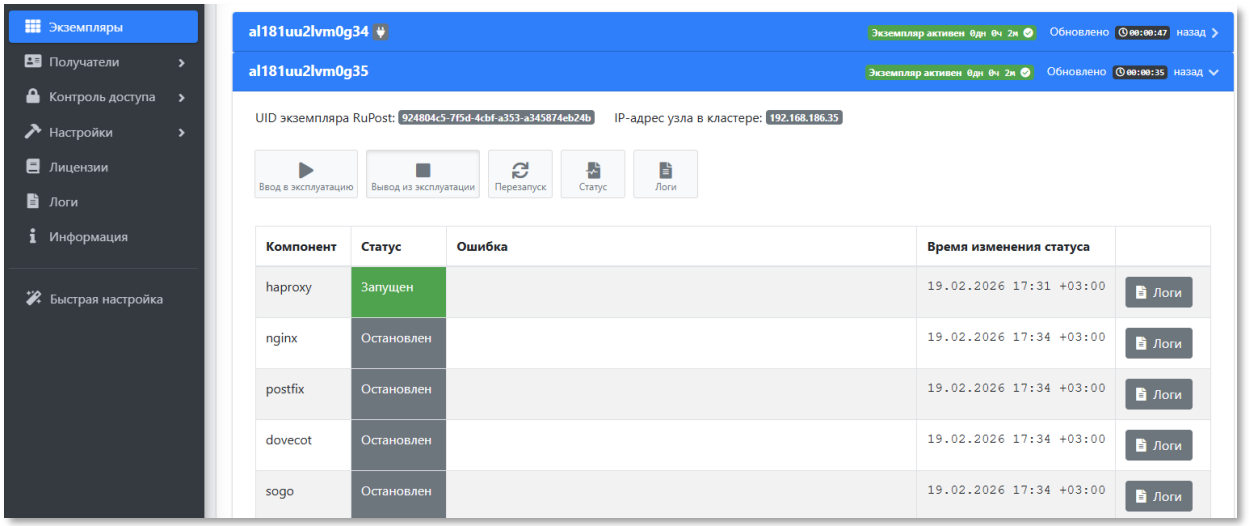
Необходимые условия:

- Доступ к консоли виртуальной машины источника и клона
  (возможность внесения изменений в конфигурацию при отключенном сетевом интерфейсе)
- Наличие в кластере минимум 2-х узлов
- Полная остановка работы RuPost на узле источнике
  (остановка работы через графический интерфейс, затем с использованием утилиты systemctl)
- Отключение сетевого интерфейса на узле источнике

Порядок действий:

1. Выбрать новый IP адрес для нового (клонируемого) узла
2. Остановить RuPost на узле источнике в графическом интерфейсе
   Экземпляры -> Вывод из эксплуатации





затем используя **systemctl** полностью остановить все модули и сервисы RuPost

```
root@al181uu2lvm0g35:~# systemctl stop rupost
root@al181uu2lvm0g35:~#
```

3. Выключить сетевой интерфейс в конфигурации узла источника
   (копии экранов приведены для варианта использования конфигурации в файле /etc/network/interfaces)

```
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*
#
auto lo
iface lo inet loopback
#
auto ens33
  iface ens33 inet static
  address 192.168.186.35
  netmask 24
  gateway 192.168.186.2
  dns-nameservers 192.168.186.12
  dns-search example.internal
```

   Поставить комментарии (#) на следующие строки:

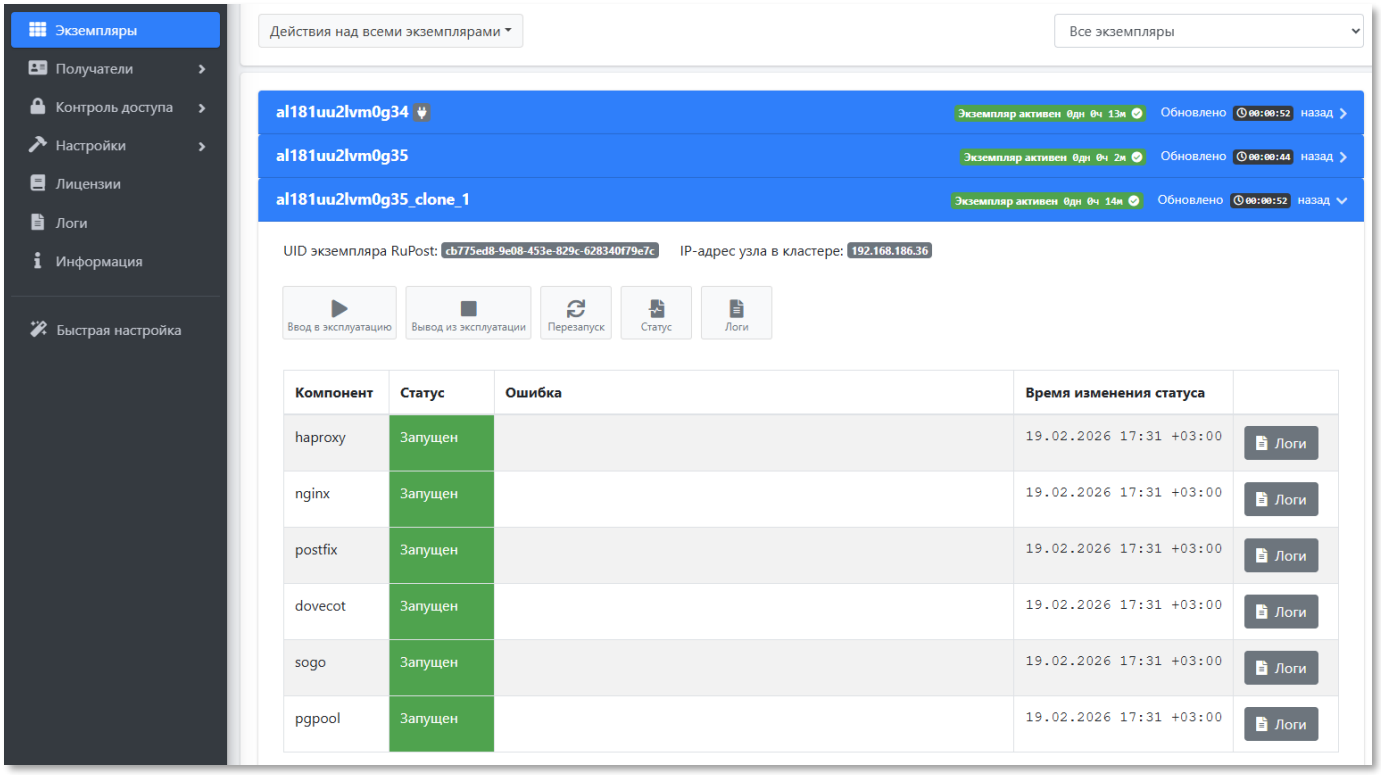
```
#auto ens33
#  iface ens33 inet static
```

4. Выключить виртуальную машину узла источника
5. Выполнить клонирование с виртуальной машины узла источника
6. Загрузить виртуальную машину узла источника, включить сетевой интерфейс
7. Перезагрузить виртуальную машину узла источника, убедиться в работе сетевого интерфейса
8. Запустить RuPost на узле источнике в графическом интерфейсе, убедиться, что RuPost работает
   Экземпляры -> Перезапуск
9. Загрузить виртуальную машину узла клона
10. Выполнить изменение IP адреса узла клона, включить сетевой интерфейс

```
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*
#
auto lo
iface lo inet loopback
#
auto ens33
  iface ens33 inet static
  address 192.168.186.36
  netmask 24
  gateway 192.168.186.2
  dns-nameservers 192.168.186.12
  dns-search example.internal
```

11. Перезагрузить узел клон, убедиться в работоспособности сетевого интерфейса и смене IP адреса

12. Запустить RuPost на узле клоне в графическом интерфейсе, убедиться, что новый узел включен в кластер



1.3.2. Ребалансировка распределения подключений IMAP по узлам кластера

В версии 3.1.0 для обеспечения равномерной нагрузки на все узлы кластера RuPost, добавлен механизм периодической ребалансировки IMAP подключений. Раз в сутки (в 02:00), происходит отключение неактивных IMAP соединений. При повторном подключении, соединение будет осуществлено на узел, имеющий наименьшее количество подключений в данный момент и, таким образом, произойдет постепенное выравнивание нагрузки по всем узлам кластера.

2. Подготовка к установке RuPost

2.1. Системные требования

В качестве платформы для системы **RuPost** может использоваться как физическое аппаратное обеспечение или "*bare metal*", так и виртуальная машина с поддержкой операционных систем семейства **GNU/Linux**.

Системные требования к инфраструктуре принципиально зависят от масштабов и профилей планируемой нагрузки.
Требования к оперативной памяти, числу ядер и производительности процессора зависят от числа подключенных пользователей и обслуживаемых почтовых ящиков и могут быть рассчитаны в соответствии с алгоритмом, описанным в **Приложении 1** данного руководства **"Расчёт системных требований в зависимости от планируемой нагрузки"**.

Внимание!

При использовании кластерного развёртывания системы или при одноузловой схеме и количестве почтовых ящиков более 1000 необходимо устанавливать PostgreSQL на отдельных выделенных для них серверах. При одноузловом развёртывании с количеством пользователей менее 1000, PostgreSQL можно устанавливать локально на тот же узел, где установлен RuPost.

2.2. Операционная система

Версия RuPost 4.0 поддерживает ОС Astra Linux Special Edition (ALSE) версий:

- 1.7.x (версии 1.7.6 и выше)
- 1.8.x (1.8.1, 1.8.2, 1.8.4 и выше)

Внимание!

В случае развертывания на версии AstraLinux 1.8.3 необходимо обратиться в службу поддержки для получения инструкций по использованию и настройке корректной версии pg_pool.

Для Astra Linux Special Edition (ALSE) 1.7 версия ядра должна быть linux-5.15-generic и выше.
Для Astra Linux Special Edition (ALSE) 1.8 версия ядра должна быть linux-6.1-generic и выше.

Для соответствующих основных версий Astra Linux необходимо использовать предназначенные для них дистрибутивы - установочные пакеты:

Операционная система	Установочный пакет RuPost
Astra Linux Special Edition 1.7.6 и выше	rupost-4.2.0-alse-amd64.run

RuPost поддерживает работу Astra Linux Special Edition в режиме защищенности «Воронеж» и «Смоленск» при выставленных по умолчанию параметров безопасности (Мандатный контроль целостности, Мандатное управление доступом, запрет трассировки ptrace, запрет пароля для команды sudo).

Перед установкой RuPost должен быть подключен расширенный репозиторий Astra Linux. При установке RuPost из данного репозитория будут установлены дополнительные пакеты:

```
lua-json lua-lpeg liblasso3 python3-tzlocal patch
```

Внимание!

При необходимости добавления сервера RuPost в домен, нужно сначала установить RuPost, а потом добавлять сервер в домен.

2.3. Синхронизация времени

Внимание!

Для корректной работы, физические серверы или виртуальные машины, на которых развернуты узлы RuPost и сопутствующие сервисы – (база данных, служба каталогов, сервис кеширования в памяти, сетевое файловое хранилище) должны быть синхронизированы по времени с допуском, не превышающим одну секунду.

Невыполнение данного требования приведет к неопределенным ошибкам функционирования системы (например, #50026), нарушению связанности кластера и целостности конфигурационных и пользовательских данных!

Также при расхождении времени на узлах не будет работать применение конфигурации к узлам кластера.

Данное требование может быть реализовано путем синхронизации времени с ближайшим расположенным, в рамках инфраструктуры, сервисом синхронизации времени на базе протоколов Network Time Protocol (NTP), Simple Network Time Protocol (SNTP) или аппаратными решениями, предоставляющие сервисы точного времени, которые используют спутниковую навигацию, данные сотовых сетей, радиосигналы, атомные часы и тому подобное.

В версии 3.2.0 в процедуру установки RuPost добавлена возможность выбора варианта настройки синхронизации времени на узлах кластера. Если синхронизация времени уже настроена, то администратор может отказаться от установки и настройки компонента синхронизации времени, используемого сервером RuPost (systemd-timesyncd).

2.4. Службы каталогов LDAP

Внимание!

Начиная с версии 3.2.0 в процедуру авторизации пользователей добавлена проверка корректности заполнения атрибута proxyAddresses. Просьба проверить, что у всех пользователей в атрибуте proxyAddresses корректно заполнена информация о первичном почтовом адресе - в формате "SMTP: e-mail@domain.ru".
При необходимости, восстановить корректные значения атрибутов LDAP для отдельного почтового ящика можно из окна свойств почтового ящика, нажав на кнопку «Сохранить».

Внимание!

Уникальность имени LDAP домена не зависит от регистра символов. Например, example.com и Example.com считаются одним и тем же доменом.
Данное правило распространяется на создание новых доменов, при этом не затронет уже заведенные домены.

Почтовая система RuPost использует сервера LDAP для авторизации пользователей. Одновременно к системе RuPost может быть подключено несколько независимых доменов LDAP. Завести почтовые ящики можно только для имеющихся активных учётных записей в службе каталогов.

Контроллер службы каталогов должен поддерживать один из способов подключения:

- протокол LDAPv3 без шифрования;
- протокол LDAPv3 с шифрованием TLS.

Добавляемый домен LDAP должен состоять в контексте имён указанных при добавлении контроллеров домена.

Правила сетевых маршрутов (route) и межсетевого экрана (firewall) должны разрешать прямое подключение всех узлов кластера RuPost к указываемым при настройке контроллерам домена на соответствующие порты (обычно 389 для протокола LDAP без шифрования и 636 для TLS LDAPS сессий).

Если контроллеры домена указываются с помощью имён, а не IP адресов, такие имена должны разрешаться в DNS или быть прописаны в файле hosts на всех узлах кластера RuPost.

В настоящее время система RuPost поддерживает следующие службы каталогов по протоколу LDAP:

- ALD Pro (версия 2.2.0 и выше)
- Avanpost DS
- FreeIPA
- Microsoft Active Directory
- Samba DC (RFC 2307)

> **Внимание!**
>
> Время ответа LDAP контроллера при авторизационном запросе от RuPost не должно превышать 10 сек. В случае высокой загрузки LDAP контроллера рекомендуется создать отдельный экземпляр LDAP контроллера для работы с RuPost.

2.4.1. Служебная учётная запись

RuPost для управления учётными записями пользователей в службе каталогов использует служебную учётную запись (сервисного пользователя). Администратор RuPost должен получить уникальные имена (DN) и пароли соответствующих сервисных аккаунтов во всех подключаемых к RuPost доменах до их подключения к RuPost. Права доступа к атрибутам и функциональное использование LDAP со стороны служебной учётной записи RuPost описано в “Приложении 3” данного Руководства.

2.4.2. Интеграция с ALD Pro

ALD Pro адаптирована для использования с RuPost, поэтому:

- не требует расширения схемы LDAP для работы с RuPost (версии ALD Pro 1.3.x и выше)
- содержит скрипт **rupostadmin** для управления сервисными учетными записями RuPost.

Внимание!

Все действия со служебной учетной записью ldapbind необходимо выполнять с соблюдением мер безопасности и не допускать сохранения паролей в истории команд. Для этого необходимо отключить запись истории вводимых команд следующей командой:
set +o history
После завершения можно восстановить запись истории команд:
set -o history

Для интеграции с RuPost в ALD Pro необходимо завести служебную учетную запись. Для этого на контролере домена ALD Pro пользователем, с возможностью использования sudo, получите Kerberos билет администратора домена. Это можно сделать, выполнив команду и введя пароль администратора:

```
set +o history

sudo kinit admin
[sudo] пароль для user:
Password for admin@DEV.INTERNAL:
```

После этого для просмотра имеющихся сервисных учётных записей системы RuPost исполните команду:

```
sudo /opt/rbta/venvs/aldpro-common/bin/python3
/opt/rbta/ad/mgmtportal/api/core/manage.py rupostadmin find
```

Если в выводе записи отсутствует:

Служебные УЗ RuPost:
ldapbind

Или необходимо создание новой записи, то для служебной записи **ldapbind** с паролем **12345678** необходимо выполнить следующее:

```
sudo /opt/rbta/venvs/aldpro-common/bin/python3
/opt/rbta/ad/mgmtportal/api/core/manage.py rupostadmin add --uid ldapbind --password
12345678
```

Для изменения пароля имеющейся записи:

```
sudo /opt/rbta/venvs/aldpro-common/bin/python3
/opt/rbta/ad/mgmtportal/api/core/manage.py rupostadmin  passwd --uid ldapbind --
password 12345678
```

Изменен пароль служебной УЗ ldapbind

В RuPost такая сервисная запись должна применяться с её полным уникальным именем (Distinguished Name, DN). Такое имя соответствует схеме:

```
uid={служебная запись RuPost},cn=sysaccounts,cn=etc,{RDN LDAP домена в 'DC' формате}
```

Так для служебной записи **ldapbind** в домене **org.example.com** уникальное имя будет следующим:

```
uid=ldapbind,cn=sysaccounts,cn=etc,dc=org,dc=example,dc=com
```

```
Управление системными учетными записями RuPost в ALDPro.

Первый позиционный аргумент определяет действие:
    "info" - справка; не требует дополнительных аргументов.
        rupostadmin info

    "find" - список существующих УЗ RuPost; не требует дополнительных аргументов.
        rupostadmin find

    "add" - создание новой УЗ RuPost; требует аргументы "Имя УЗ" и "Пароль УЗ".
        rupostadmin add -u user -p password

    "passwd" - изменение пароля существующей УЗ RuPost; требует аргументы "Имя УЗ" и "Пароль УЗ".
                    rupostadmin passwd -u user -p password

    "del" - удаление существующей УЗ RuPost; требует один или несколько аргументов "Имя УЗ".
                    rupostadmin passwd -u user
                  rupostadmin passwd -u user1 -u user2 -u user3

Именные аргументы:
-u [UID], --uid [UID]     Имя УЗ RuPost: не может быть пустым или "new",
                          может содержать цифры, латиницу в нижнем регистре и символы "-", "_", "$", "."
-p [PASSWORD], --password [PASSWORD]  Пароль УЗ RuPost: может быть от 7 до 255 символов,
                                      не может начинаться или заканчиваться пробелом,
                                      не может содержать символы "`", "~", "{", "}", """, '"'
```

В состав ALD Pro входит служба DNS. В случае если RuPost использует DNS из ALD Pro необходимо внести настройки п. 2.10 данного руководства в конфигурацию службы DNS ALD Pro. Это можно сделать двумя вариантами, с использованием графического интерфейса администратора ALD Pro, или выполнить настройки из командной строки, , пользователем с правами администратора, получив Kerberos билет администратора. Более подробно о назначении и использовании записей DNS для работы RuPost можно прочитать в п. 2.10 данного Руководства.

Ниже приведены примеры настроек для обоих вариантов.

В примерах используются следующие параметры:

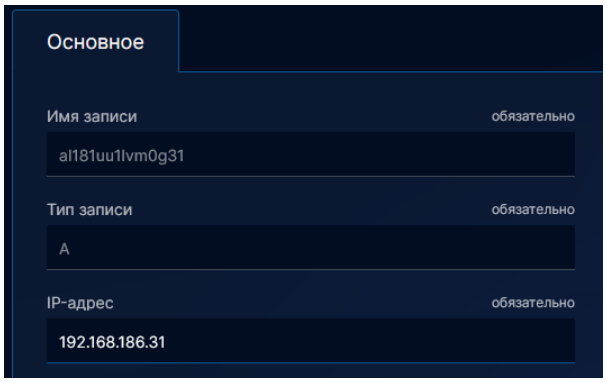
Имя домена: `example.dn`

Полное имя узла (FQDN) RuPost: `al181uu1lvm0g31.example.dn`

IP адрес узла RuPost: `192.168.186.31`

Вариант ввода настроек в графическом интерфейсе администратора ALD Pro.

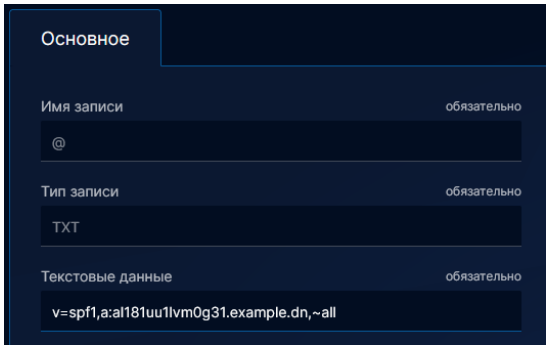
Внесение записи типа A



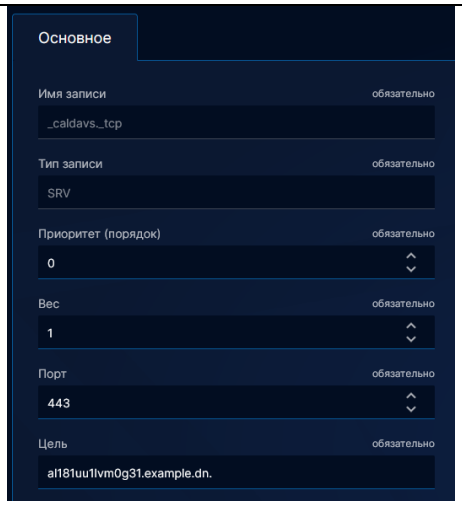
Внесение записи типа PTR

Внесение записи типа MX

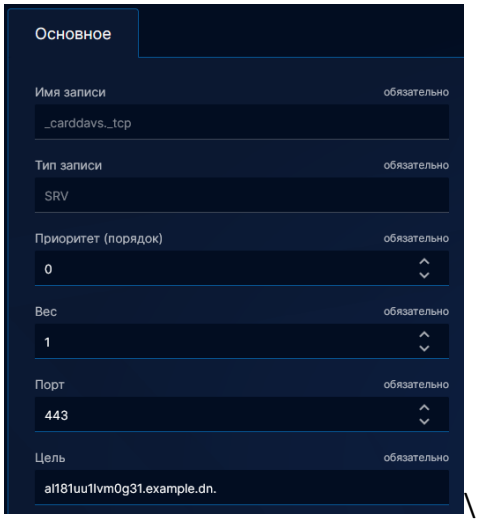
Внесение записи типа TXT



Внесение записи типа _caldavs._tcp

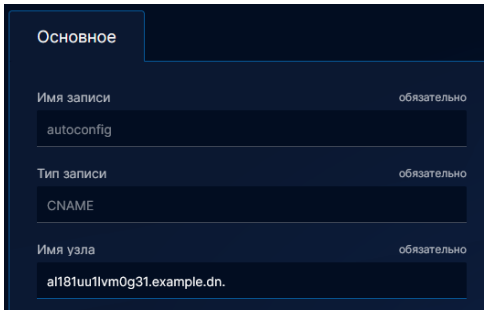


Внесение записи типа _carddavs._tcp



\

Внесение записи типа autoconfig



Внесение записи типа autodiscover

Вариант ввода настроек из командной строки с получением Kerberos билета администратора ALD Pro.

Получение Kerberos билета и ввод пароля администратора ALD Pro пользователем с правами администратора.

```
kinit admin
Password for admin@EXAMPLE.DN: <пароль_администратора_ALD_Pro>
```

Внесение записи типа A

```
ipa dnsrecord-add example.dn al181uu1lvm0g31 --a-rec="192.168.186.31"
```

Внесение записи типа PTR

```
ipa dnsrecord-add 186.168.192.in-addr.arpa. 31 --ptr-rec=al181uu1lvm0g31.example.dn.
```

Внесение записи типа MX

```
ipa dnsrecord-add example.dn @ --mx-rec="10 al181uu1lvm0g31.example.dn."
```

Внесение записи типа TXT

```
ipa dnsrecord-add example.dn @ --txt-rec="v=spf1,a:al181uu1lvm0g31.example.dn,~all"
```

Внесение записи типа _caldavs._tcp

```
ipa dnsrecord-add example.dn _caldavs._tcp --srv-rec="0 1 443 al181uu1lvm0g31.example.dn."
```

Внесение записи типа _carddavs._tcp

```
ipa dnsrecord-add example.dn _carddavs._tcp --srv-rec="0 1 443 al181uu1lvm0g31.example.dn."
```

Внесение записи типа autoconfig

```
ipa dnsrecord-add example.dn autoconfig --cname-rec="al181uu1lvm0g31.example.dn."
```

Внесение записи типа autodiscover

```
ipa dnsrecord-add example.dn autodiscover --cname-rec="al181uu1lvm0g31.example.dn."
```

Выполнить проверку внесенных записей можно следующим образом.

```
ipa dnsrecord-show example.dn al181uu1lvm0g31

  Имя записи: al181uu1lvm0g31
  A record: 192.168.186.31

ipa dnsrecord-show 186.168.192.in-addr.arpa. 31

  Имя записи: 31
  PTR record: al181uu1lvm0g31.example.dn.

ipa dnsrecord-show example.dn @

  Имя записи: @
  MX record: 10 al181uu1lvm0g31.example.dn.
  NS record: al175m2g11.example.dn.
  TXT record: v=spf1,a:al181uu1lvm0g31.example.dn,~all
```

```
ipa dnsrecord-show example.dn _caldavs._tcp

  Имя записи: _caldavs._tcp
  SRV record: 0 1 443 al181uu1lvm0g31.example.dn.

ipa dnsrecord-show example.dn _carddavs._tcp

  Имя записи: _carddavs._tcp
  SRV record: 0 1 443 al181uu1lvm0g31.example.dn.

ipa dnsrecord-show example.dn autoconfig

  Имя записи: autoconfig
  CNAME record: al181uu1lvm0g31.example.dn.

ipa dnsrecord-show example.dn autodiscover

  Имя записи: autodiscover
  CNAME record: al181uu1lvm0g31.example.dn.
```

Внести изменения в записи можно с помощью команды **ipa dnsrecord-mod <изменяемый_параметр>**

Для удаления записи используется команда **ipa dnsrecord-del <удаляемый_параметр>**

Пример использования возможностей DNS BIND9, входящей в состав ALD Pro, для балансировки входящих сообщений между несколькими узлами RuPost.

В данном примере рассматривается вариант конфигурации DNS BIND9 с 3 узлами RuPost, IP адреса:
192.168.186.31
192.168.186.32
192.168.186.33

Домен:
example.internal

Наименование почтового сервера:
mail.example.internal

Контроллер ALD Pro, IP адрес:
192.168.186.12

Конфигурация выполняется на контроллере ALD Pro, пользователем с правами администратора. В начале необходимо получить Kerberos билет и ввести пароль администратора ALD Pro:

```
kinit admin
Password for admin@EXAMPLE.INTERNAL: <пароль_администратора_ALD_Pro>
```

Далее производятся необходимые настройки:

```
ipa dnsrecord-add example.internal mail --a-rec="192.168.186.31"
ipa dnsrecord-add example.internal mail --a-rec="192.168.186.32"
ipa dnsrecord-add example.internal mail --a-rec="192.168.186.33"
ipa dnsrecord-add 186.168.192.in-addr.arpa. 31 --ptr-rec=mail.example.internal.
ipa dnsrecord-add 186.168.192.in-addr.arpa. 32 --ptr-rec=mail.example.internal.
ipa dnsrecord-add 186.168.192.in-addr.arpa. 33 --ptr-rec=mail.example.internal.
ipa dnsrecord-add example.internal @ --mx-rec="10 mail.example.internal."
ipa dnsrecord-add example.internal @ --txt-rec="v=spf1,a:mail.example.internal,~all"
ipa dnsrecord-add example.internal _caldavs._tcp --srv-rec="0 1 443 mail.example.internal."
ipa dnsrecord-add example.internal _carddavs._tcp --srv-rec="0 1 443 mail.example.internal."
ipa dnsrecord-add example.internal autoconfig --cname-rec="mail.example.internal."
```

```
ipa dnsrecord-add example.internal autodiscover --cname-rec="mail.example.internal."
```

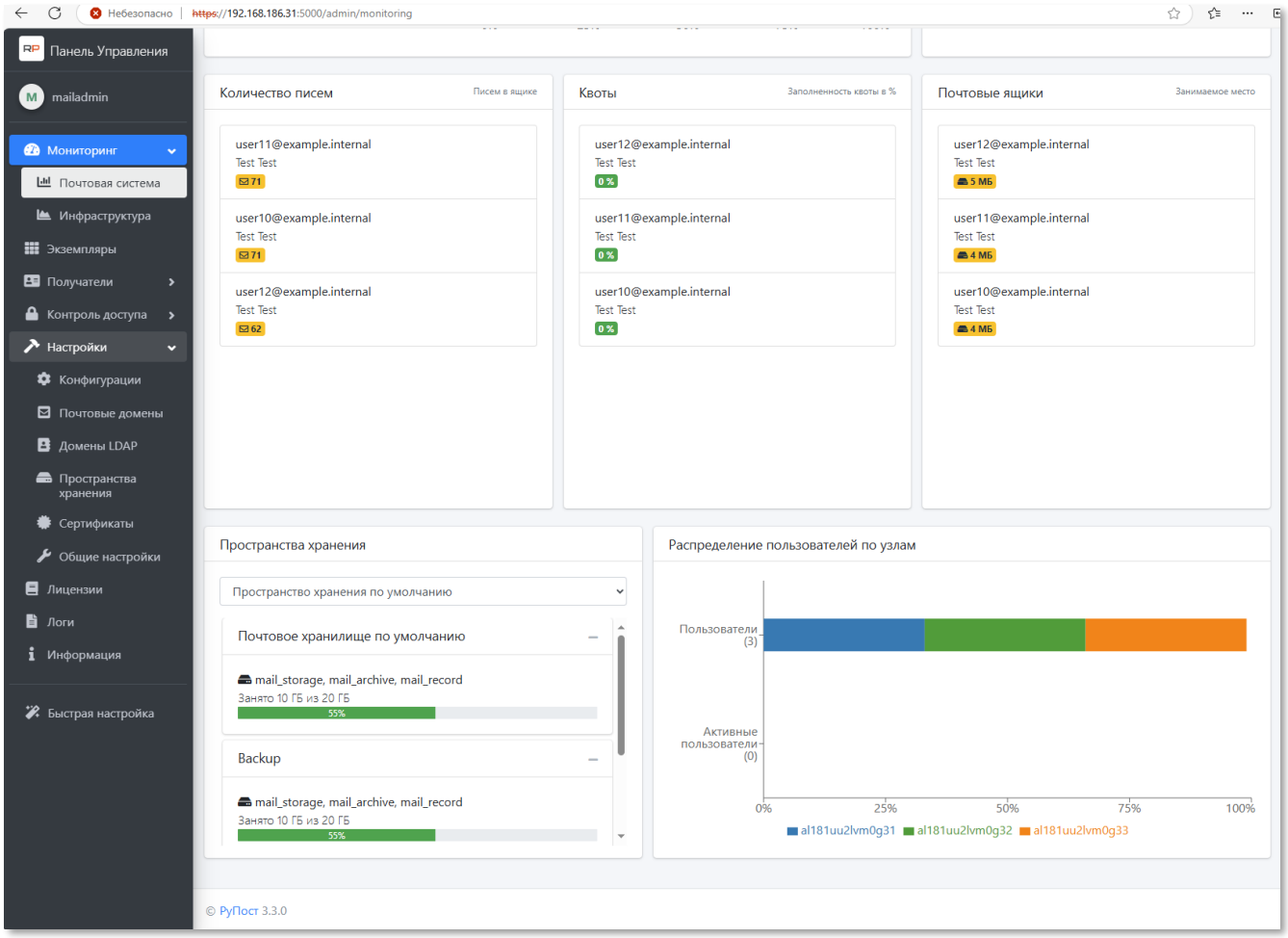
Проверку можно выполнить с любого узла RuPost, необходимо убедиться, что контроллер ALD Pro указан в качестве сервера службы DNS.

```
cat /etc/resolv.conf
search example.internal
nameserver 192.168.186.12
```

и выполнить проверку:

```
for i in {1..10}; do dig +short mail.example.internal; echo; done
192.168.186.31
192.168.186.32
192.168.186.33

192.168.186.32
192.168.186.31
192.168.186.33

192.168.186.33
192.168.186.32
192.168.186.31

192.168.186.33
192.168.186.31
192.168.186.32

192.168.186.33
192.168.186.32
192.168.186.31

192.168.186.32
192.168.186.31
192.168.186.33

192.168.186.31
192.168.186.33
192.168.186.32

192.168.186.31
192.168.186.32
192.168.186.33

192.168.186.33
192.168.186.32
192.168.186.31

192.168.186.31
192.168.186.33
192.168.186.32
```

При использовании генератора входящего потока сообщений для контроля можно использовать графический интерфейс администратора RuPost, экран «Мониторинг», «Почтовая система», «Распределение пользователей по узлам»:



Таким образом можно использовать возможности BIND9 для балансировки входящих сообщений почтового кластера.

2.4.3. Подготовка FreeIPA

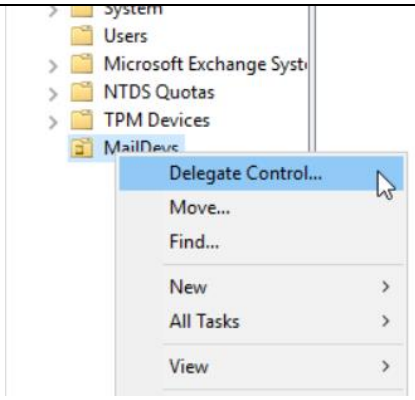
Для подготовки FreeIPA на мастер контроллере домена требуется выполнить bash сценарий, поставляемый вендором по соответствующему запросу.

Внимание!

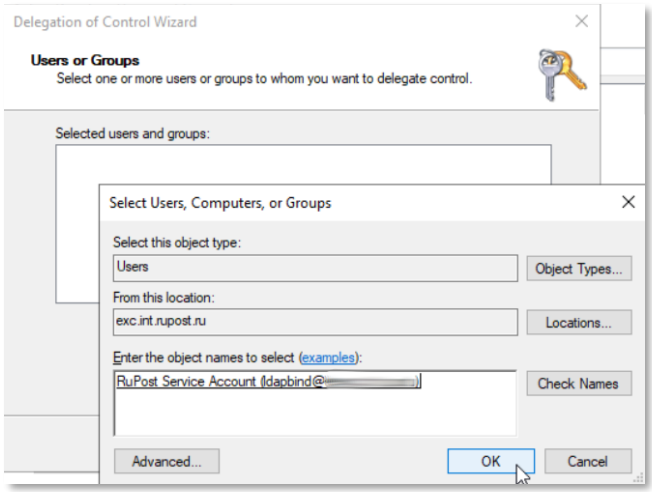
В службе каталогов FreeIPA атрибут **mail** является MULTI-VALUE, т.е. может содержать несколько значений. Для работы RuPost необходимо, чтобы в атрибуте **mail** содержался только один адрес.

2.4.4. Подготовка Microsoft Active Directory

После заведения в службе каталогов сервисной учётной записи, посредством которой RuPost будет управлять пользовательскими атрибутами своих клиентов, необходимо делегировать упомянутой учётной записи соответствующие права. Для этого выберите в службе каталогов подразделение, которое выделено для обслуживания в RuPost, и в контекстном меню выберите Delegate Control...

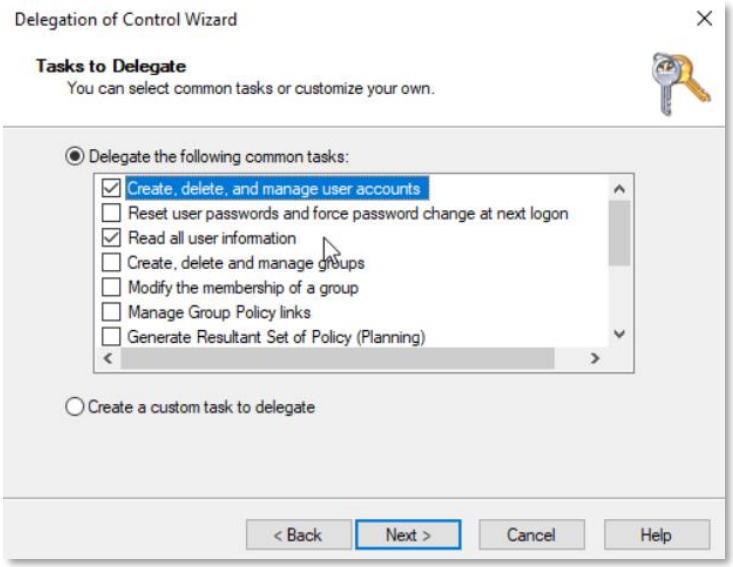


Далее необходимо добавить сервисную учётную запись RuPost для расширения прав.



После этого выберете в списке:

- Create, delete, and manage user accounts
- Read all user information



Сохраните выбранные привилегии. Теперь служебная учётная запись RuPost будет обладать достаточными правами для управления пользовательскими атрибутами в выбранном подразделении.

2.4.5. Поддержка Samba DC

Для подготовки Samba DC на мастер-контроллере домена требуется выполнить специальный bash-сценарий, поставляемый вендором по запросу. Указанный сценарий выполняется на стороне контроллера домена и предназначен для подготовки схемы и объектов каталога, необходимых для интеграции с RuPost.

После создания в службе каталогов сервисной учётной записи, посредством которой RuPost будет управлять пользовательскими атрибутами своих клиентов, необходимо делегировать данной учётной записи соответствующие права доступа.

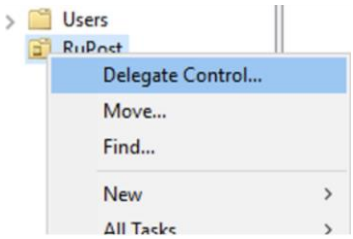
Делегирование прав выполняется аналогично процедуре, применяемой в Microsoft Active Directory.

Для этого необходимо открыть оснастку Active Directory Users and Computers от имени пользователя с административными правами домена, например, выполнив следующую команду:

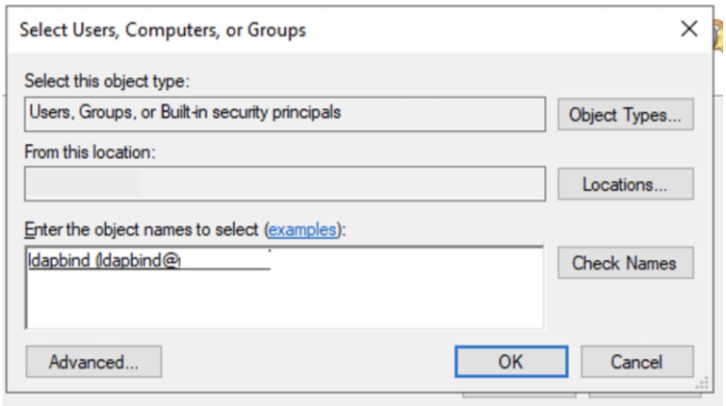
runas /netonly /user:administrator@example.com "mmc dsa.msc /server=dc.example.com"

После ввода пароля будет открыта оснастка Active Directory Users and Computers, подключённая к указанному контроллеру домена.

Далее в службе каталогов необходимо выбрать подразделение (OU), выделенное для обслуживания пользователей RuPost, и в контекстном меню выбрать пункт Delegate Control….

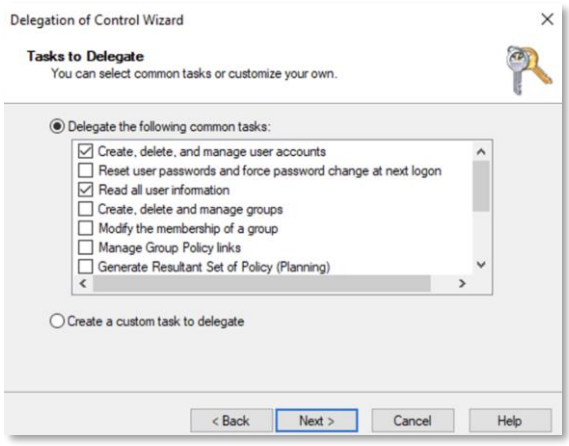


На следующем шаге следует добавить сервисную учётную запись RuPost, для которой требуется расширение прав доступа.



В списке разрешений необходимо выбрать следующие пункты:

- Create, delete, and manage user accounts
- Read all user information



После завершения мастера делегирования сохраните выбранные привилегии.
В результате выполненных действий сервисная учётная запись RuPost будет обладать достаточными правами.
Набор поддерживаемых LDAP атрибутов совпадает с атрибутами LDAP домена Active Directory и соответствует RFC 2307.

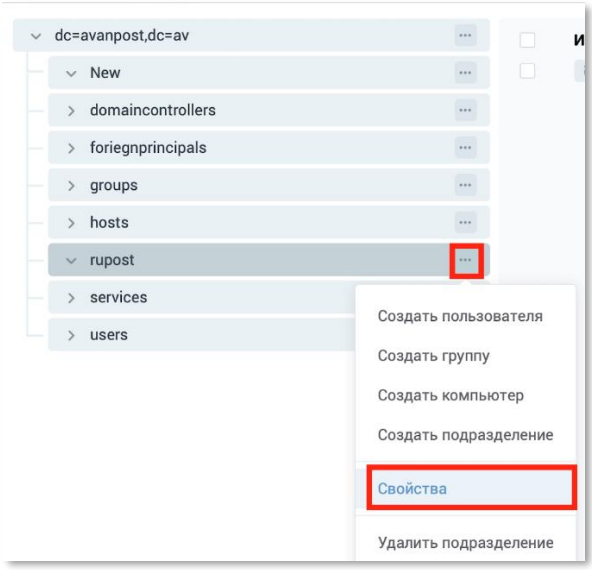
2.4.6. Подготовка Avanpost DS

Для интеграции RuPost со службой каталогов Avanpost DS необходимо создать сервисную учётную запись, посредством которой RuPost будет управлять пользовательскими атрибутами.

Для Avanpost DS версии 1.9.0-20 и ниже.

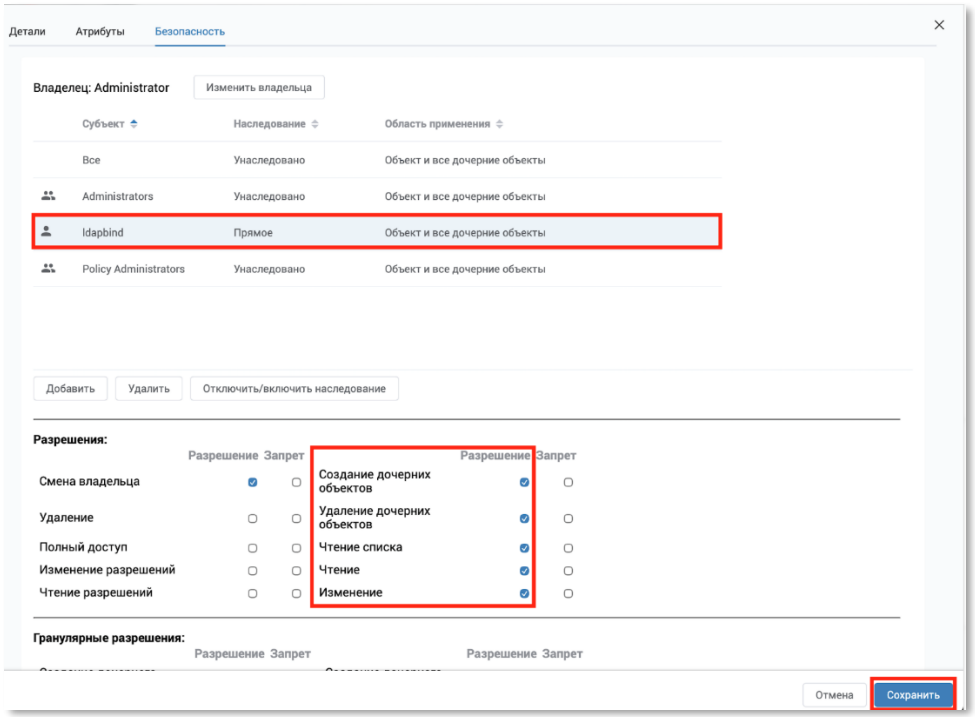
Сервисная учётная запись RuPost должна быть включена в группу Administrators.

Для Avanpost DS версии 1.9.1-21 для интеграции необходимо создать сервисную учётную запись и выдать ей права на необходимую OU.



На вкладке безопасность добавить сервисную УЗ с областью применения на этот объект и все дочерние. И выдать необходимые права:

- Создание дочерних объектов
- Удаление дочерних объектов
- Чтение списка
- Чтение
- Изменение



2.5. Система управления базами данных

Поддерживаемые СУБД:

- PostgreSQL версии не ниже 11 (**рекомендуется версия 15 и выше)**
- Tantor
- Кластер PostgreSQL на технологии Patroni

По умолчанию для конфигурации на одном узле предлагается использовать СУБД PostgreSQL, выполняющуюся на самом узле RuPost. В этом случае, необходимые базы данных будут созданы автоматически во время установки.

Внимание!

Пароли от баз данных хранятся в открытом виде в конфигурационных файлах системы, поэтому необходимо предпринять организационно-технические меры, нацеленные на ограничение и мониторинг действий круга лиц имеющих доступ к этим файлам как на просмотр, так и на редактирование - для этого рекомендуется использовать встроенные механизмы ОС Astra Linux.

Внимание!

Для корректной работы почтовой системы RuPost необходима настройка СУБД на работу с кодировкой ru_RU.UTF-8.
В конфигурационный файл СУБД postgresql.conf внести следующие настройки:

```
lc_messages = 'ru_RU.UTF-8'                     # locale for system error message
lc_monetary = 'ru_RU.UTF-8'                     # locale for monetary formatting
lc_numeric = 'ru_RU.UTF-8'                      # locale for number formatting
lc_time = 'ru_RU.UTF-8'                         # locale for time formatting
```

Внимание!

В процессе работы почтовой системы RuPost открываются соединения к СУБД. Необходимо рассчитать требуемое количество соединений по рекомендациям в п. 10.5 и увеличить в конфигурационный файле СУБД postgresql.conf параметр **max_connections** на это значение для варианта отдельного сервера СУБД. Пример увеличения количества подключений к СУБД для варианта кластера patroni приведен ниже.

2.5.1. Настройка Tantor BE/SE

Для работы с новой установкой базы данных Tantor необходимо провести предварительную настройку.

Устанавливаем ожидание подключения на всех TCP/IP адресах (по умолчанию это только "localhost"). IPv6 сюда тоже входит - если это нежелательно, то можно выставить "0.0.0.0":

```
sudo -iu postgres psql -c "ALTER SYSTEM SET listen_addresses = '*'"
```

Задаём пароль для пользователя postgres (ввод отображаться не будет):

```
sudo -iu postgres psql -c "\password"
```

Настраиваем pg_hba.conf для разрешения подключения с любого адреса к любой базе данных любым пользователем по паролю:

```
sudo bash -c 'echo "host all all all scram-sha-256" >> /var/lib/postgresql/tantor-se-15/data/pg_hba.conf'
```

Перезапускаем сервис чтобы внесённые изменения вступили в силу

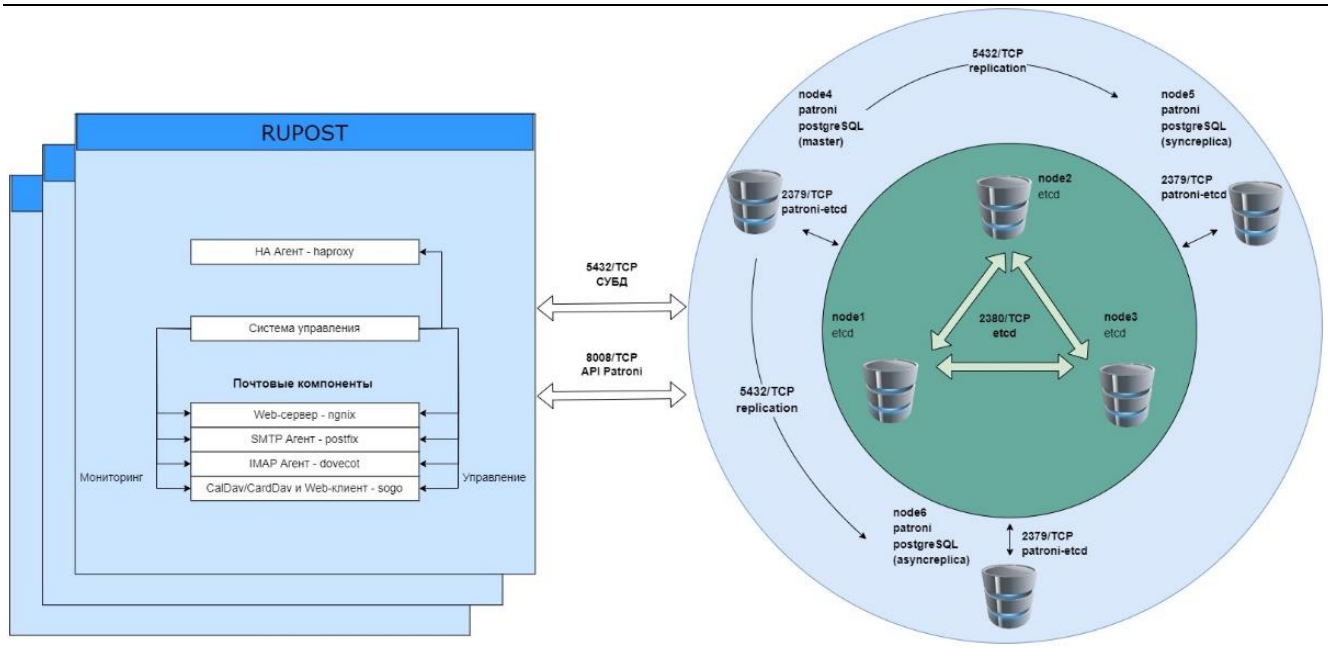
```
sudo systemctl restart tantor-se-server-15
```

Затем выполняем настройки, указанные в п. 2.5.2.

2.5.2. Общие настройки для PostgreSQL, Tantor BE/SE, кластера на основе patroni

Для повышения надежности работы RuPost, в версии 3.0 добавлена поддержка кластера баз данных PostgreSQL, построенном с использованием технологии Patroni.

Диаграмма архитектуры взаимодействия компонентов для кластера Patroni из трех узлов СУБД PostgreSQL:



В состав кластера Patroni входят несколько узлов (СУБД), один из которых является основным (master), а остальные – репликами (replica). Данные основного узла реплицируются на все реплики и, таким образом, при сбое на основном узле возможно переключение на реплику (назначение одной из реплик основным узлом).

Особенностью кластера Patroni является автоматическое переключение на другой узел кластера при обнаружении сбоя в работе основного узла. RuPost подключается к API Patroni, получает информацию о том, какой узел является основным и переключается на работу с этим узлом.

В случае, если требуется подключение к серверу управления база данных (СУБД), развёрнутому в инфраструктуре организации, то необходимо до начала установки RuPost выполнить следующие действия:

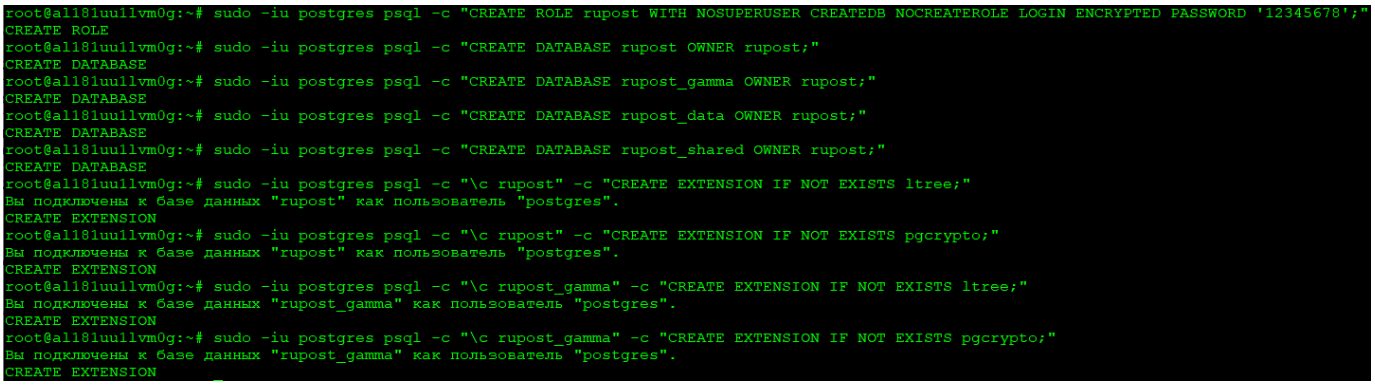
1. Войти на сервер управления база данных (СУБД) пользователем ОС с правами root.
2. Создать в СУБД пользователя rupost с определенными правами (`NOSUPERUSER CREATEDB NOCREATEROLE LOGIN`) и назначить ему пароль.
3. Создать базы данных (`rupost rupost_gamma rupost_data rupost_shared`), необходимые для работы сервера RuPost.
4. Установить расширения (`ltree pgcrypto`) реализующие дополнительные функции для баз данных `rupost и rupost_gamma`.

Все действия должны выполняться пользователем ОС postgres с правами «суперпользователя» в СУБД на сервере БД. Если используется кластер patroni данные действия выполняются на узле с ролью «Лидер» (Leader).

Пример создания пользователя rupost с паролем 12345678, баз данных и расширений на сервере СУБД PostgreSQL.

```
sudo -iu postgres psql -c "CREATE ROLE rupost WITH NOSUPERUSER CREATEDB NOCREATEROLE
LOGIN ENCRYPTED PASSWORD '12345678';"
```

```
sudo -iu postgres psql -c "CREATE DATABASE rupost OWNER rupost;"
sudo -iu postgres psql -c "CREATE DATABASE rupost_gamma OWNER rupost;"
sudo -iu postgres psql -c "CREATE DATABASE rupost_data OWNER rupost;"
sudo -iu postgres psql -c "CREATE DATABASE rupost_shared OWNER rupost;"
sudo -iu postgres psql -c "\c rupost" -c "CREATE EXTENSION IF NOT EXISTS ltree;"
sudo -iu postgres psql -c "\c rupost" -c "CREATE EXTENSION IF NOT EXISTS pgcrypto;"
sudo -iu postgres psql -c "\c rupost_gamma" -c "CREATE EXTENSION IF NOT EXISTS
ltree;"
sudo -iu postgres psql -c "\c rupost_gamma" -c "CREATE EXTENSION IF NOT EXISTS
pgcrypto;"
```



Другой вариант выполнения этих действий создать файл rp.sql и выполнить его.

```
sudo -iu postgres nano rp.sql
```

Содержимое файла rp.sql

```
CREATE ROLE rupost WITH NOSUPERUSER CREATEDB NOCREATEROLE LOGIN ENCRYPTED PASSWORD
'12345678';
CREATE DATABASE rupost OWNER rupost;
CREATE DATABASE rupost_gamma OWNER rupost;
CREATE DATABASE rupost_data OWNER rupost;
CREATE DATABASE rupost_shared OWNER rupost;
\c rupost
CREATE EXTENSION IF NOT EXISTS ltree;
CREATE EXTENSION IF NOT EXISTS pgcrypto;
\c rupost_gamma
CREATE EXTENSION IF NOT EXISTS ltree;
CREATE EXTENSION IF NOT EXISTS pgcrypto;
\c rupost_shared
CREATE EXTENSION IF NOT EXISTS ltree;
CREATE EXTENSION IF NOT EXISTS pgcrypto;
```

Выполнение файла на сервере СУБД

```
sudo -iu postgres psql -f rp.sql
```

```
root@al181uu11vm0g:~# sudo -iu postgres psql -f rp.sql
CREATE ROLE
CREATE DATABASE
CREATE DATABASE
CREATE DATABASE
CREATE DATABASE
Вы подключены к базе данных "rupost" как пользователь "postgres".
CREATE EXTENSION
CREATE EXTENSION
Вы подключены к базе данных "rupost_gamma" как пользователь "postgres".
CREATE EXTENSION
CREATE EXTENSION
Вы подключены к базе данных "rupost_shared" как пользователь "postgres".
CREATE EXTENSION
CREATE EXTENSION
```

Проверка создания баз данных и расширений

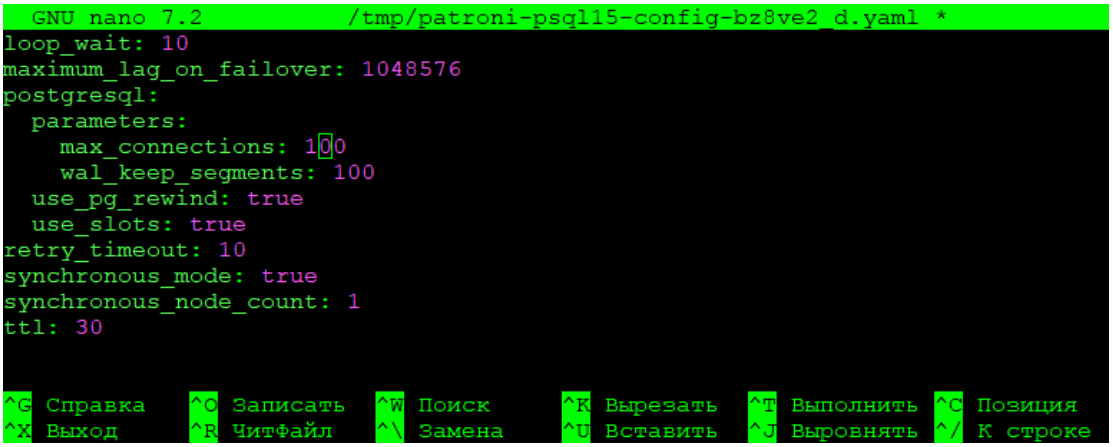
```
sudo -iu postgres psql -c "\du" -c "\l" -c "\c rupost" -c "\dx" -c "\c rupost_gamma"
-c "\dx"
```

```
root@al181uu11vm0g:~# sudo -iu postgres psql -c "\du" -c "\l" -c "\c rupost" -c "\dx" -c "\c rupost_gamma" -c "\dx"
                                   Список ролей
 Имя роли |                                Атрибуты                                 | Член ролей
----------+-------------------------------------------------------------------------+------------
 postgres | Суперпользователь, Создаёт роли, Создаёт БД, Репликация, Пропускать RLS | {}
 rupost   | Создаёт БД                                                              | {}

                                                  Список баз данных
     Имя      |     Владелец      | Кодировка | LC_COLLATE  |  LC_CTYPE   | локаль ICU | Провайдер локали |     Права доступа
--------------+-------------------+-----------+-------------+-------------+------------+------------------+-----------------------
 postgres     | postgres          | UTF8      | ru_RU.UTF-8 | ru_RU.UTF-8 |            | libc             |
 rupost       | rupost            | UTF8      | ru_RU.UTF-8 | ru_RU.UTF-8 |            | libc             |
 rupost_data  | rupost            | UTF8      | ru_RU.UTF-8 | ru_RU.UTF-8 |            | libc             |
 rupost_gamma | rupost            | UTF8      | ru_RU.UTF-8 | ru_RU.UTF-8 |            | libc             |
 rupost_shared | rupost           | UTF8      | ru_RU.UTF-8 | ru_RU.UTF-8 |            | libc             |
 template0    | postgres          | UTF8      | ru_RU.UTF-8 | ru_RU.UTF-8 |            | libc             | =c/postgres          +
              |                   |           |             |             |            |                  | postgres=CTc/postgres
 template1    | postgres          | UTF8      | ru_RU.UTF-8 | ru_RU.UTF-8 |            | libc             | =c/postgres          +
              |                   |           |             |             |            |                  | postgres=CTc/postgres
 template_rbac | pg_database_admin | UTF8      | ru_RU.UTF-8 | ru_RU.UTF-8 |            | libc             |
(8 строк)

Вы подключены к базе данных "rupost" как пользователь "postgres".
                  Список установленных расширений
  Имя   | Версия |   Схема    |                   Описание
--------+--------+------------+-----------------------------------------------
 ltree  | 1.2    | public     | data type for hierarchical tree-like structures
 pgcrypto | 1.3  | public     | cryptographic functions
 plpgsql | 1.0   | pg_catalog | PL/pgSQL procedural language
(3 строки)

Вы подключены к базе данных "rupost_gamma" как пользователь "postgres".
                  Список установленных расширений
  Имя   | Версия |   Схема    |                   Описание
--------+--------+------------+-----------------------------------------------
 ltree  | 1.2    | public     | data type for hierarchical tree-like structures
 pgcrypto | 1.3  | public     | cryptographic functions
 plpgsql | 1.0   | pg_catalog | PL/pgSQL procedural language
(3 строки)
```

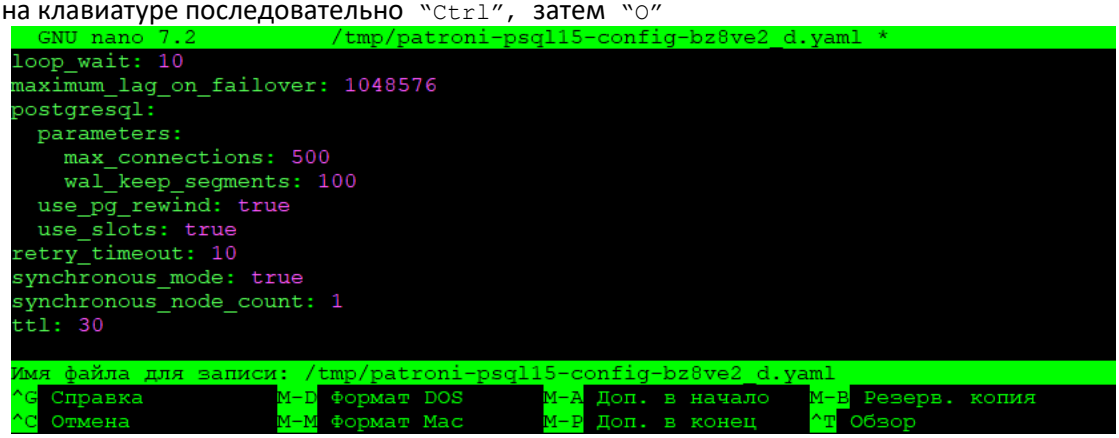
Необходимое для работы RuPost количество подключений к СУБД PostgreSQL рассчитывается по формуле, указанной в п.10.5, пример расчета приведен в п. 10.6. Настройка количества подключений к СУБД PostgeSQL производится в файле конфигурации postgresql.conf для отдельного СУБД PostgreSQL. В случае если СУБД PostgreSQL используется в составе кластера patroni изменение конфигурации кластера производится с помощью утилиты patronictl. При этом в редакторе будет открыт актуальный файл с конфигурацией, который необходимо отредактировать и сохранить.

Пример получения данных о количестве подключений (100) и изменение количества подключений на 500 в кластере patroni. Команды могут быть выполнены на любом из узлов кластера patroni.

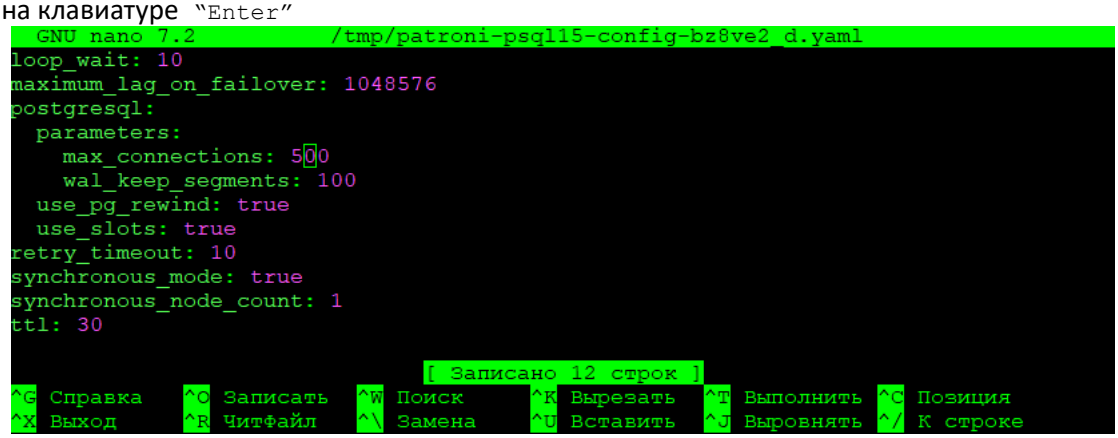
```
sudo -iu postgres psql -p 5000 -c "\x auto" -c "SHOW max_connections"
Расширенный вывод применяется автоматически.
 max_connections
-----------------
 100
(1 строка)

sudo patronictl -c /etc/patroni/config.yml edit-config
```



Нажмите на клавиатуре последовательно “Ctrl”, затем “O”



Нажмите на клавиатуре “Enter”



Нажмите на клавиатуре последовательно "Ctrl", затем "X"

```
---
+++
@@ -2,6 +2,7 @@
 maximum_lag_on_failover: 1048576
 postgresql:
   parameters:
+    max_connections: 500
     wal_keep_segments: 100
   use_pg_rewind: true
   use_slots: true
```

Подтвердите вносимые изменения нажав на клавиатуре “y”

```
Apply these changes? [y/N]: y
Configuration changed
```

Для вступления в силу выполненных изменений необходимо перезапустить кластер patroni.

Внимание!

Выполнение данной команды приведет к временному прекращению работы пользователей и приложений с кластером, при этом существующие подключения будут разорваны!

```
patronictl -c /etc/patroni/config.yml restart patroni-psql15
```

```
+ Cluster: patroni-psql15 --------------+--------------+---------+----+-----------+-----------------+
| Member          | Host                | Role         | State   | TL | Lag in MB | Pending restart |
+-----------------+---------------------+--------------+---------+----+-----------+-----------------+
| al181uu1lvm0g67 | 192.168.186.67:5000 | Leader       | running |  6 |           | *               |
| al181uu1lvm0g68 | 192.168.186.68:5000 | Sync Standby | running |  6 |         0 | *               |
| al181uu1lvm0g69 | 192.168.186.69:5000 | Replica      | running |  6 |         0 | *               |
+-----------------+---------------------+--------------+---------+----+-----------+-----------------+
When should the restart take place (e.g. 2025-01-26T15:24)  [now]:
Are you sure you want to restart members al181uu1lvm0g67, al181uu1lvm0g68, al181uu1lvm0g69?
```

Подтвердите перезапуск кластера нажав на клавиатуре “y”

```
[y/N]: y
Restart if the PostgreSQL version is less than provided (e.g. 9.5.2)  []:
Success: restart on member al181uu1lvm0g67
Success: restart on member al181uu1lvm0g68
Success: restart on member al181uu1lvm0g69
```

Кластер успешно перезапущен, проверьте актуальное значение количества соединений

```
sudo -iu postgres psql -p 5000 -c "\x auto" -c "SHOW max_connections"
Расширенный вывод применяется автоматически.
 max_connections
-----------------
 500
(1 строка)
```

Рекомендуется

Параметры настройки СУБД PostgrSQL для оптимального быстродействия (файл postgresql.conf):

```
shared_buffers = 25% от объема оперативной памяти, выделяемого для работы СУБД PostgreSQL
temp_buffers = 256MB
work_mem = 64MB
effective_io_concurrency = 1
max_worker_processes = 64
wal_buffers = 16MB
wal_writer_delay = 2000ms
wal_sync_method = open_sync
synchronous_commit = off
max_wal_size = 256MB
idle_in_transaction_session_timeout = 300000
hot_standby_feedback = off
```

Внимание!

При использовании СУБД PostgreSQL в составе кластера patroni параметр:
`hot_standby_feedback`

при смене лидера, по умолчанию, устанавливается в состояние
`on`

это приводит к удержанию слотами репликации информации об удаленных записях в таблицах, быстрому заполнению таблиц, увеличению времени работы запросов и отклика СУБД.

Необходимо для всех узлов СУБД PostgreSQL кластера patroni установить значение параметра:
`hot_standby_feedback = off`

Внимание!

При использовании СУБД PostgreSQL на ОС Astra Linux Special Edition (ALSE) с уровнем защищенности Воронеж или Смоленск необходимо предоставить системному пользователю postgres права на чтение из системной БД сведений о метках безопасности и привилегиях пользователей:

```
sudo usermod -a -G shadow postgres
sudo setfacl -d -m u:postgres:r /etc/parsec/macdb
sudo setfacl -R -m u:postgres:r /etc/parsec/macdb
sudo setfacl -m u:postgres:rx /etc/parsec/macdb
sudo setfacl -d -m u:postgres:r /etc/parsec/capdb
sudo setfacl -R -m u:postgres:r /etc/parsec/capdb
sudo setfacl -m u:postgres:rx /etc/parsec/capdb
```

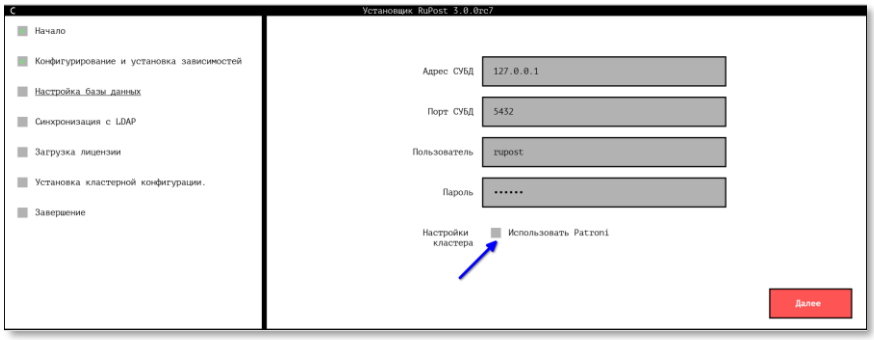
Создать группу и пользователя rupost

```
sudo groupadd -g 420 rupost
sudo useradd -u 420 -g 420 -s /usr/sbin/nologin -d /home/rupost -m rupost
```

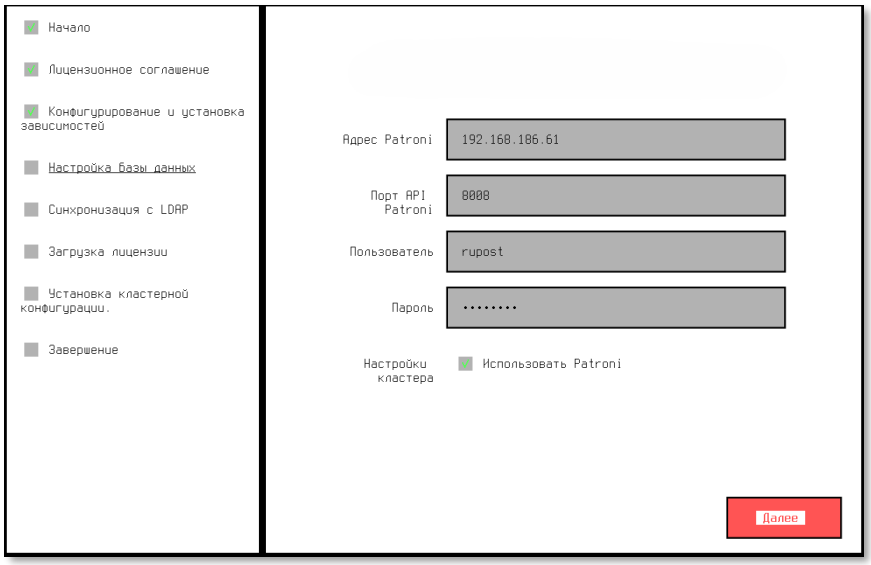
Задать уровень конфиденциальности для пользователя rupost:

```
sudo pdpl-user rupost -l 0:0
```

При установке RuPost с использованием кластера Patroni, в установщике на шаге конфигурирования подключения к базе данных, необходимо выбрать опцию “Использовать Patroni” и указать адрес узла кластера баз данных. Достаточно указать адрес только одного узла - после этого RuPost получит адреса остальных узлов кластера непосредственно из Patroni.



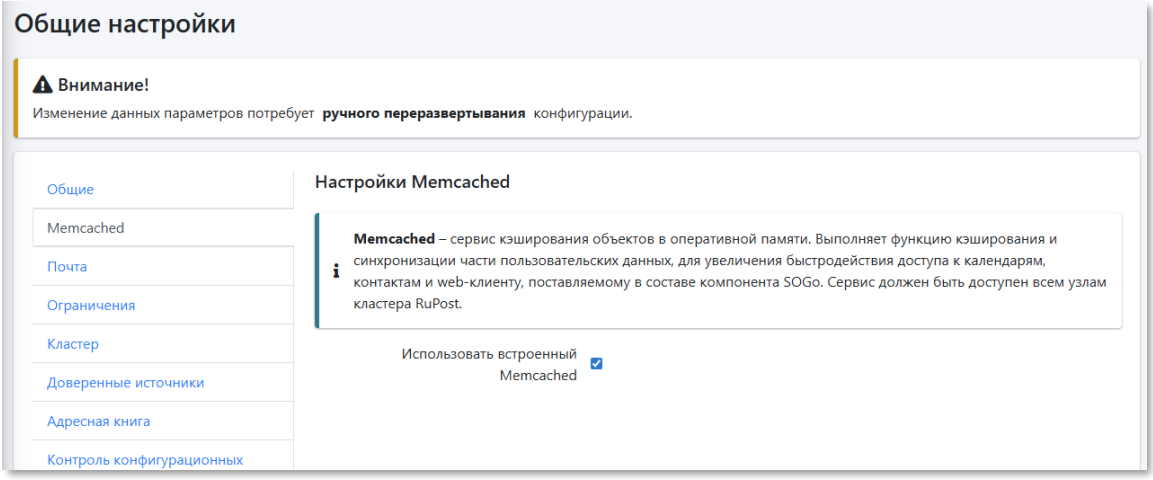
В открывшемся окне необходимо указать IP адрес узла кластера patroni, порт REST API patroni, имя пользователя и пароль.



2.6. Служба кэширования объектов в оперативной памяти rupost-cs (Memcached)

Поддерживается служба Memcached версии не ниже 1.4.33.

Начиная с версии 4.2.0 установка выделенного сервера Memcached – не обязательна, можно использовать Memcached, который устанавливается на узлах кластера в ходе установки RuPost.



В случае кластерной конфигурации, при использовании встроенного Memcached, этот модуль устанавливается на все узлы кластеру RuPost и в случае выхода из строя узла с активным Memcached происходит переключение на другой работающий узел. Таким образом обеспечивается резервирование Memcached для узлов кластера RuPost.

Внимание!

В памяти сервиса Memcached хранятся пары логин/пароль от базы данных, и хеш паролей пользователей от ldap, которые можно получить, используя команды **telnet** или **netcat**.
В случае кластерной конфигурации, необходимо ограничить подключения к сервису всем, кроме узлов RuPost с помощью сетевых средств защиты (например, межсетевого экрана).
При установке RuPost на одном узле, служба **Memcached** обслуживает только локальные подключения, но нужно обеспечить установку корректных прав доступа для пользователей, имеющих доступ к узлу.

Размер выделяемой памяти рассчитывается по формуле:

```
Требуемая память (килобайт) = 512 * (количество доменов LDAP) + (10 * количество пользователей)
```

Требуемая память указывается в **Мб** конфигурационном файле `/etc/memcached.conf`.

```
# set ram size to 8MBytes to 256MBytes
CACHESIZE="4096"
```

Указать требуемую память можно также через командный интерфейс Memcahed:

```
memcached -m 3072
```

Ключ -m задает значение объема памяти в Мб. Например:

```
-m 64
```

означает 64 Мб.
После внесения изменений нужно выполнить перезапуск сервиса memcached командой:

```
service memcached restart
```

2.7. Пространства хранения, группы ящиков и хранилища

Для обеспечения сценариев высокой доступности RuPost, в версии 3.0 добавлены новые средства управления хранением почты – **Пространство хранения** (MailSpace), **Группа ящиков** (MailBox Group) и **Хранилище** (MailStore).

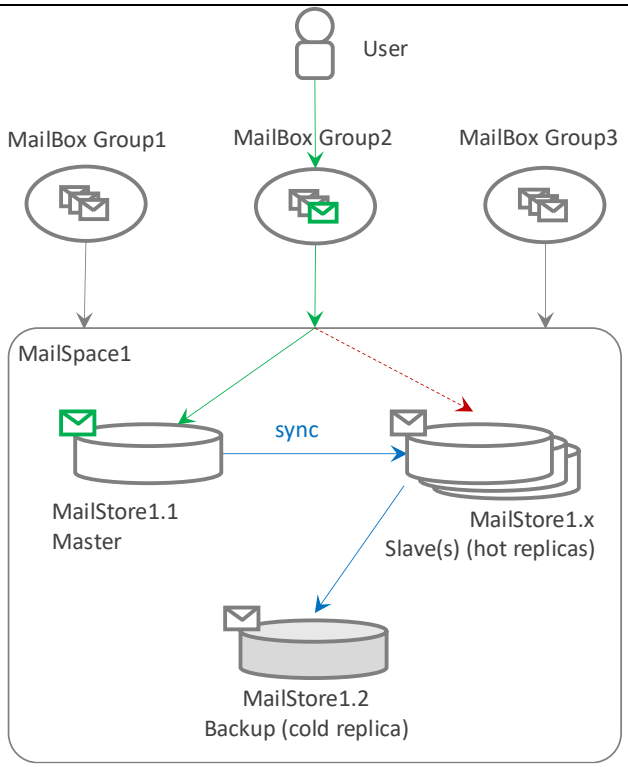
Пространство хранения (MailSpace) - совокупность нескольких хранилищ почты (MailStore), связанных правилами репликации. Минимально необходимо наличие хотя бы одного хранилища почты. Хранилища почты делятся по типам - одно из них является мастером (активное, обслуживает почту в данный момент), несколько хранилищ могут быть ведомыми (slave, "горячие" реплики мастер-хранилища) и, кроме того, может быть одно резервное (Backup, "холодная" реплика) хранилище. Состояние всех slave и backup хранилищ почты постоянно синхронизируется посредством периодической односторонней репликации в направлении мастер -> slave / backup. Slave хранилища почты считаются "горячими", т.е. при сбое на мастер-хранилище возможно переключение на slave.

Группа ящиков (MailBox Group) - это набор почтовых ящиков, обслуживаемых одним Пространством хранения (MailSpace). Все ящики, входящие в одну Группу ящиков (MailBox Group) расположены в том Пространстве хранения (MailSpace), которое указано в свойствах этой Группы ящиков. Одно Пространство хранения может быть использовано для хранения нескольких Групп ящиков. Каждый Почтовый ящик (MailBox) принадлежит только одной Группе ящиков.

Хранилище почты (MailStore) - набор точек монтирования. Минимально необходимо наличие одной точки монтирования для хранения почтовых файлов в формате Maildir. В том случае, когда в Общих настройках системы установлено, что должны использоваться Архивы и/или Record Storage, то в свойствах хранилища должны быть указаны точки монтирования для Архивов и/или Record Storage соответственно. Точки монтирования являются уникальными для всех master и slave хранилищ всех Пространств хранения. Уникальность отслеживается по полному пути - адрес NFS сервера + имя папки.

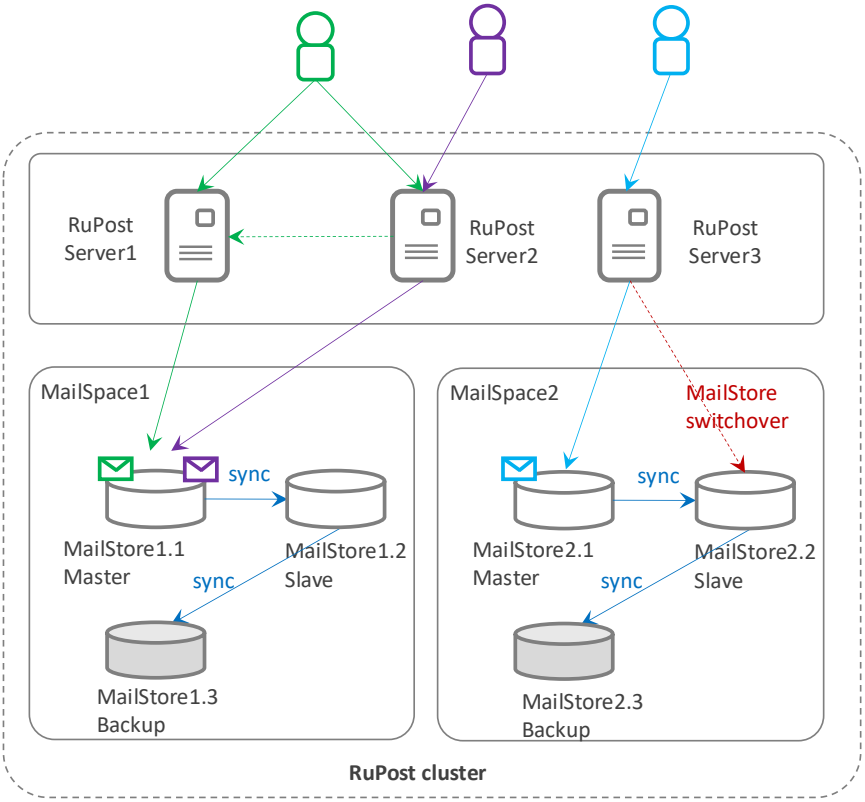
Резервное хранилище (Backup, "холодная" реплика). В пространство хранения может быть добавлено Backup-хранилище ("холодная" реплика), которое используется как источник данных для Системы резервного копирования (СРК). Периодичность синхронизации Backup-хранилища, в общем случае, имеет гораздо больший интервал (т.е. больше "отстает" от мастер-хранилища) чем у slave-хранилищ ("горячих" реплик). Backup-хранилище не может быть назначено мастером, т.е. на него нельзя переключить обслуживание почты.

Логические связи между сущностями, которыми оперирует новое поколение RuPost 3, отражены на диаграмме:



Настройка Групп почтовых ящиков, Пространств хранения и Хранилищ осуществляется как из Панели управления RuPost так и с помощью командного интерфейса (CLI).

Общая логика коммуникаций в кластере RuPost 3.0 представлена на диаграмме:



Уникальным преимуществом представленной архитектуры RuPost является возможность работы разных пользователей с почтовыми ящиками, размещенными в одном пространстве хранения, через разные узлы почтового кластера RuPost (серверы RuPost). Такой подход существенно повышает надежность работы кластера в целом и качественно оптимизирует нагрузку на отдельные серверы обработки почты в кластере – что повышает эффективность использования инфраструктурных ресурсов и производительность почтовой системы.

Для обеспечения перехода от единого почтового хранилища (до версии 3.0) к использованию Пространств хранения, при установке RuPost создаются:

- **Пространство хранения** – “Пространство хранения по умолчанию”
- **Группа ящиков** – “Группа ящиков по умолчанию”.
- **Хранилище** – “Хранилище по умолчанию”.

Все почтовые ящики, созданные ранее, после миграции, находятся в группе ящиков по умолчанию, которая хранится в Пространстве по умолчанию.

Внимание!

В версии RuPost 4.0.0 мастер хранилище, архив и скрытая копия RecordStorage реализованы в виде отдельных самостоятельных хранилищ, раздел 2.8.1 «Рекомендации по настройке сетевых файловых хранилищ».

2.8. Подключение сетевых каталогов файловой системы NFSv4

Внимание!
Необходимо использование NFS не ниже версии 4.

Внимание!
В версии 3.0 индексные файлы перенесены внутрь папки почтового ящика (Maildir) так что отдельная точка монтирования для индексов не требуется. Соответственно, при обновлении до версии 3.0, индексные файлы будут перемещены внутрь папок MailDir при выполнении миграции в ходе установки RuPost.

Внимание!
Время переноса индексных файлов определяется количеством почтовых ящиков, а также количеством папок в каждом ящике. При установке RuPost этап миграции может занять существенное время. Информация о ходе процесса миграции индексных файлов доступна в monitor.log.

Внимание!
Запрещено на сервере NFS экспортировать каталоги файловой системы, имеющие иерархическую вложенность относительно уже экспортированных родительских каталогов. Например, в файловой системе сервера NFS каталог recordstorage не должен располагаться внутри хранилища писем.

Ниже представлен рекомендуемый пример организации структуры каталогов для пространства хранения.

Мастер-хранилище почты MailStore (каталог **master**), "горячая" реплика мастер-хранилища (каталог **replica**) и "холодная" реплика мастер-хранилища для системы резервного копирования (каталог **backup**). Каталоги **queuesprim** и **queuessec** используются для почтовых очередей**.** В каталог **archives** перемещаются сообщения электронной почты, предназначенные для длительного хранения. В каталог **recordstorage** перемещаются удаленные пользователем почтовые сообщения.

```
/srv/
└── ms01
    ├── backup
    │   ├── arc
    │   ├── mb
    │   └── rstor
    ├── master
    │   ├── arc
    │   ├── mailboxes
    │   └── rstor
    ├── queprim
    ├── quesec
    └── replica
        ├── arc
        ├── mailboxes
        └── rstor
```

Для подключения сетевых файловых хранилища и каталога почтовых очередей необходимо экспортировать сетевые каталоги NFSv4 со следующими настройками:

- Для всех подключаемых каталогов (почтовых очередей, хранилища почтовых ящиков, пользовательских архивов и управления записями “record storage”) необходимо активировать параметры **rw, sync, no_subtree_check, no_root_squash** (как правило, в файле /etc/exports).

Также для каждого подключаемого сетевого каталога, например, `/srv/nfs/MailStorage`, на стороне сервера NFS необходимо назначить **UID:GID** равные **420:420** соответственно. Сделать это можно, по аналогии выполнив команду на сервере NFS для всех подключаемых каталогов **(за исключением каталога с почтовыми очередями)**:

```
sudo chown 420:420 -R /srv/nfs/MailStorage
```

Для каталога почтовых очередей владельца необходимо назначить по схеме **UID:GID** равным **421:root** соответственно без рекурсивного назначения прав **(без ключа -R)**.

```
sudo chown 421:root /srv/nfs/MailQueues
```

Внимание!

Простое копирование каталогов и содержимого NFS между хранилищами без корректного контроля прав и параметров приведет к неработоспособности системы.

Пример экспорта каталогов конфигурации NFS:

```
/srv/nfs/QueuesPrim 10.154.22.0/24(rw,sync,no_subtree_check,no_root_squash)
/srv/nfs/MailStorage 10.154.22.0/24(rw,sync,no_subtree_check,no_root_squash)
/srv/nfs/MailArchive 10.154.22.0/24(rw,sync,no_subtree_check,no_root_squash)
/srv/nfs/MailRecord 10.154.22.0/24(rw,sync,no_subtree_check,no_root_squash)

/srv/nfs/QueuesSec 10.154.22.0/24(rw,sync,no_subtree_check,no_root_squash)
```

*[где 10.154.22.0/24 — пример подсети, в которой расположены узлы RuPost]

** [`/srv/nfs/QueuesSec` экспорт каталога с резервного сервера NFS для очередей]

2.8.1. Рекомендации по настройке сетевых файловых хранилищ

Внимание!

Для получения наилучшего результата настоятельно рекомендуется выполнить проектирование топологии и структуры организации данных корпоративной почтовой системы RuPost до установки и конфигурирования.

В версии 4.0 были значительно расширены возможности организации данных и структурирования дискового пространства. Предоставлены широкие возможности распределения данных почтовых ящиков в зависимости от требований к отказоустойчивости и безопасности, а также для соответствия требованиям регуляторов, оптимизация информационных потоков в сети при доступе к данным и распределения больших объемов данных по устройствам хранения.

При проектировании топологии и схемы размещения данных появились дополнительные возможности оптимально распределить как нагрузку от сетевых потоков, так и значительные объемы данных (сотни ТБ) между компонентами инфраструктуры, эффективно использовать вычислительные ресурсы.

В рамках Пространства хранения поддерживаются три типа хранилищ, каждое из которых может быть индивидуально сконфигурировано:

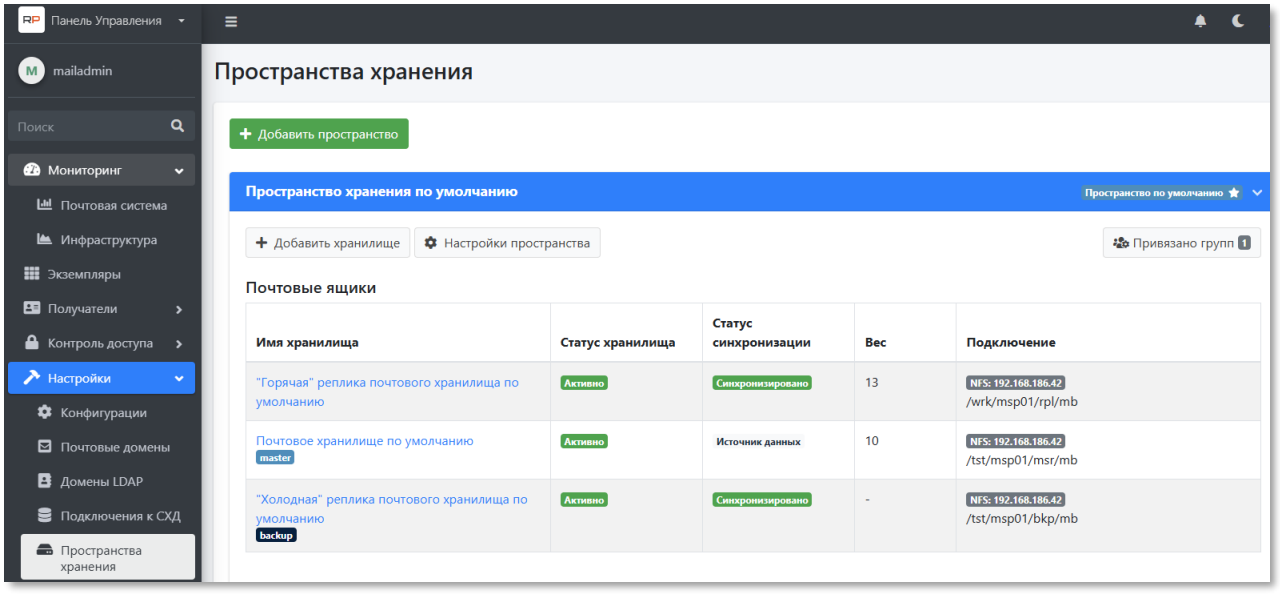
- Почта
- Архивы
- Скрытая копия (RecordStorage)

Типы, роли и размещение хранилищ версии RuPost 4.0 в рамках одного Пространства хранения

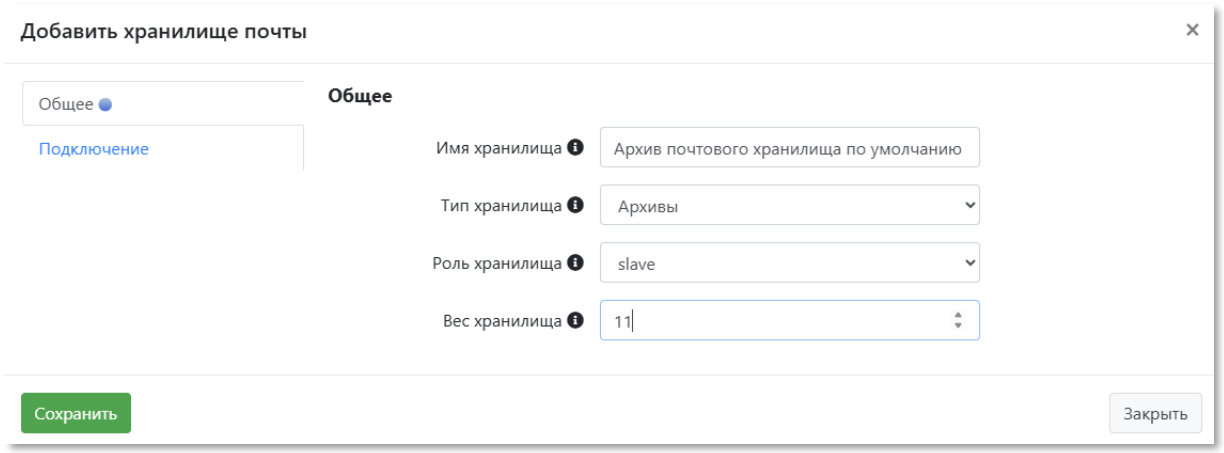
назначение	тип	роль	реплики		кол-во хранилищ
			«горячая»	«холодная»	
Основное хранилище, в него помещаются получаемые сообщения	Почтовые ящики	master	✓	✓	1
сообщения из основного хранилища, ежедневно перемещаемые согласно Политике архивирования писем	Архивы	master	✓	✓	1
удаленные пользователем сообщения (необходимо при наличии требований от регуляторов)	RecordStorage	master	✓	✓	1
«горячая» реплика основного хранилища, постоянно синхронизируется	Почтовые ящики	slave	✗	✗	несколько
«горячая» реплика Архивов, постоянно синхронизируется	Архивы	slave	✗	✗	несколько
«горячая» реплика скрытой копии Record Storage, постоянно синхронизируется	RecordStorage	slave	✗	✗	несколько
«холодная» реплика основного хранилища, источником является «горячая» реплика, при ее отсутствии - основное хранилище, синхронизируется по расписанию или заданию времени начала синхронизации	Почтовые ящики	backup	✗	✗	1
«холодная» реплика Архивов, источником является «горячая» реплика, при ее отсутствии - основное хранилище, синхронизируется по расписанию или заданию времени начала синхронизации	Архивы	backup	✗	✗	1
«холодная» реплика RecordStorage, источником является «горячая» реплика, при ее отсутствии - основное хранилище, синхронизируется по расписанию или заданию времени начала синхронизации	RecordStorage	backup	✗	✗	1

Пример создания хранилища типа «Архивы»

Для создания хранилища типа Архив, необходимо в Панели управления выбрать раздел «Настройки», в нем выбрать «Пространства хранения».



В открывшемся окне нажать кнопку «Добавить хранилище». В поле «Имя хранилища» задать название хранилища, в поле «Тип хранилища» из выпадающего списка выбрать «Архивы» и указать значение веса хранилища.



Внимание!

При первичном создании хранилища, роль хранилища следует оставить «slave». Перед созданием выполняется проверка и, в случае отсутствия хранилища данного типа с ролью «мастер», вновь создаваемому хранилищу будет назначена роль «мастер» автоматически.

Примеры организации хранения данных в зависимости от типа/размера предприятия и требований

Приведенные ниже примеры конфигурации хранилищ были составлены на основе следующих критериев:

- **Малое предприятие** (приоритет – оптимизация затрат на инфраструктуру, нет требований от регуляторов).
- **Среднее предприятие – вариант 1** (базовые требования к отказоустойчивости, нет требований от регуляторов).
- **Среднее предприятие - вариант 2** (повышенные требования к отказоустойчивости, требования регуляторов).
- **Крупное предприятие** (высокие требования к отказоустойчивости, требования регуляторов).

Малое предприятие

В данном варианте основным приоритетом является оптимизация и сокращение используемых ресурсов. Относительно невысокие требования надежности и отказоустойчивости. В случае аварии допускается некоторый объем потери данных за временной промежуток от момента возникновения аварийной ситуации до времени создания крайней резервной копии. Определена временная глубина хранения данных, свыше которой данные удаляются. В случае выбора минимального варианта (основное хранилище и «холодная реплика») рекомендуется размещение «холодной» реплики на отдельном ресурсе инфраструктуры от основного хранилища, при формировании процедур резервного копирования предлагается использовать различные варианты сжатия данных, которые обеспечат уменьшение хранимых данных в несколько раз.

Рекомендации по организации данных:

- единственное пространство хранения
- мастер хранилище
- «холодная» реплика мастер хранилища
- Архив

Общий объем хранимых данных равен двукратному объему данных мастер хранилища плюс % заложенный на Архив. Выполняется регулярное резервное копирование «холодной» реплики. При возникновении аварии работа почтовой системы останавливается, производится восстановление данных из крайней резервной копии. Данные перемещаются в Архив через временной промежуток, равный глубине хранения. Для Архива периодически используется процедура очистки.

Среднее предприятие (вариант 1)

Одним из приоритетов являются требования к отказоустойчивости. В данном варианте приоритетом является баланс между оптимизацией и отказоустойчивостью ресурсов. В зависимости от объема данных возможно некоторое отставание «горячей» реплики от основного хранилища и небольшая потеря данных. Определена временная глубина хранения данных, свыше которой данные удаляются. Рекомендуется размещение «горячей» и «холодных» реплик на отдельных от основного хранилища ресурсах инфраструктуры, использование системы резервного копирования.

Рекомендации по организации данных:

- единственное пространство хранения
- мастер хранилище
- «горячая» реплика мастер хранилища
- «холодная» реплика мастер хранилища
- Архив

Общий объем хранимых данных равен трехкратному объему данных мастер хранилища плюс % заложенный на Архив. При аварии на основном хранилище, возможен переход на «горячую» реплику. Выполняется регулярное резервное копирование «холодной» реплики. Данные перемещаются в Архив через временной промежуток, равный глубине хранения. Для Архива периодически используется процедура очистки.

Среднее предприятие (вариант 2)

Определяющим является набор требований отказоустойчивости и регуляторов. В зависимости от объема данных, возможно некоторое отставание «горячей» реплики от основного хранилища и небольшая потеря данных при выходе из строя мастер-хранилища. Временная глубина хранения данных определяется требованиями регуляторов.

Рекомендации по организации данных:

- Несколько пространств хранения (количество зависит от объема данных и требований регуляторов)

В каждом Пространстве хранения:

- Мастер-хранилище
- «горячая(ие)» реплика(и) мастер-хранилища
- «холодная» реплика
- скрытая копия - RecordStorage
- «горячая» реплика(и) скрытой копии
- «холодная» реплика скрытой копии
- Архив
- «горячая» реплика архива
- «холодная» реплика архива

Общий объем хранимых данных равен суммарному объему хранилищ. При аварии на основном хранилище возможен переход на «горячую» реплику. Выполняется регулярное резервное копирование «холодной» реплики. Данные перемещаются в Архив через временной промежуток, равный глубине хранения по требованиям регуляторов. Очистка Архива производится в соответствии с требованиями регуляторов к временной глубине хранения.

Крупное предприятие

В данном варианте предъявляются высокие требования к отказоустойчивости. Рекомендуется проектирование топологии и схемы для эффективного использования возможности RuPost по структурированию и распределению данных между пространствами хранения и хранилищами.

Используется совмещение механизмов репликации RuPost и репликации средствами систем хранения данных. Для создания резервных копий используется высокопроизводительная, распределенная система резервного копирования. В случае использовании синхронной блочной репликации средствами систем хранения данных минимизирована потеря данных. Временная глубина хранения данных определяется требованиями регуляторов.

Рекомендации по организации данных:

- пространства хранения (количество зависит от объема данных и требований регуляторов)

В каждом пространстве хранения:

- мастер хранилище
- «горячая(ие)» реплика(и) мастер хранилища
- реплики мастер хранилищ средствами СХД (систем хранения данных)
- «холодная» реплика мастер хранилища
- скрытая копия RecordStorage
- «горячая» реплика скрытой копии
- «холодная» реплика скрытой копии
- Архив
- «горячая» реплика архива
- «холодная» реплика архива

Общий объем хранимых данных равен суммарному объему хранилищ. При аварии на основном хранилище возможен переход на «горячую» реплику. Выполняется регулярное резервное копирование «холодной» реплики. Данные перемещаются в Архив через временной промежуток, равный глубине хранения по требованиям регуляторов. Очистка Архива производится в соответствии с требованиями регуляторов к временной глубине хранения.

Внимание!

При первоначальной установке корпоративной почтовой системы RuPost автоматически создается локальное основное хранилище на дисковом пространстве сервера RuPost.

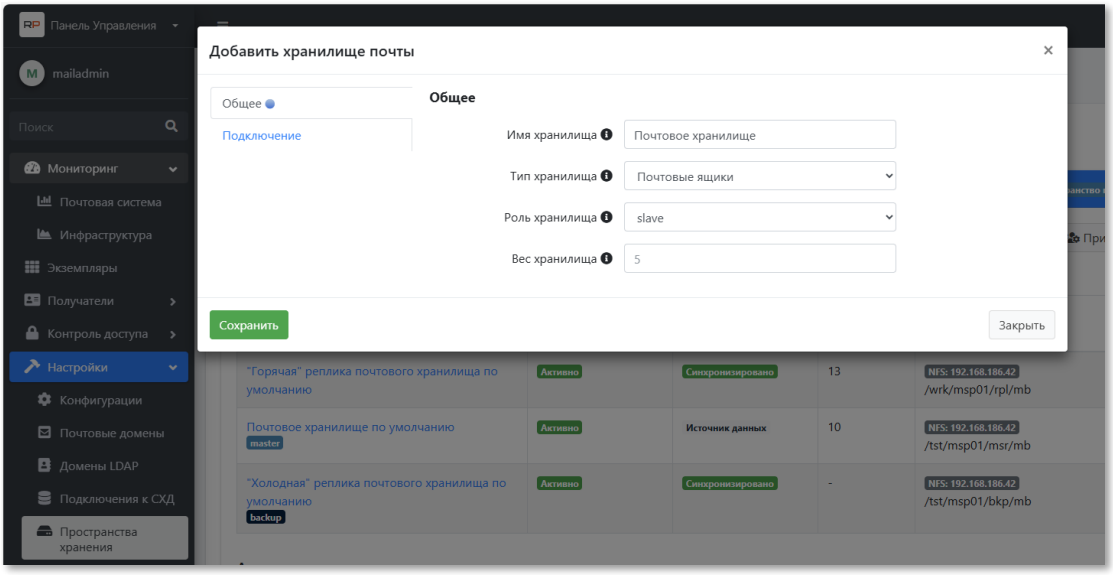
Вариант локального основного хранилища предназначен только для одиночного сервера и не может быть использован для кластерной установки.

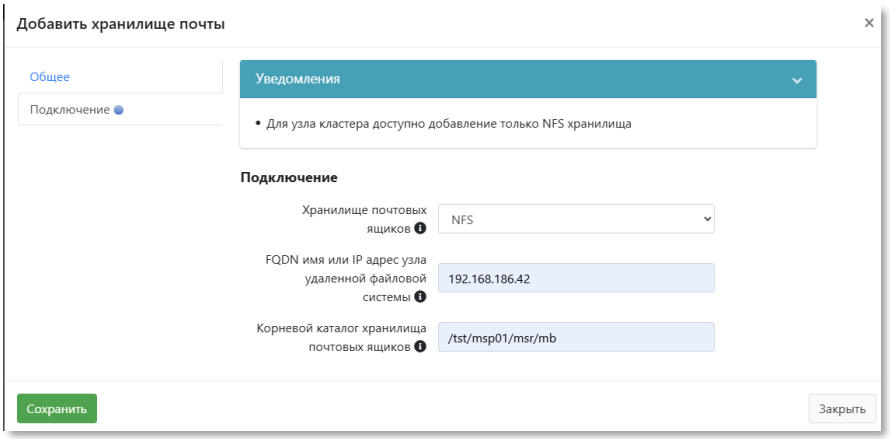
Действия по замене локального хранилища на NFS хранилище по умолчанию для развертывания кластера

По умолчанию, при первоначальной установке, основное хранилище создается на локальном диске сервера. В варианте кластерной установки, при наличии основного хранилища доступного на ресурсе по протоколу NFS, после завершения первичного сохранения конфигурации необходимо выполнить следующие действия в Панели управления:

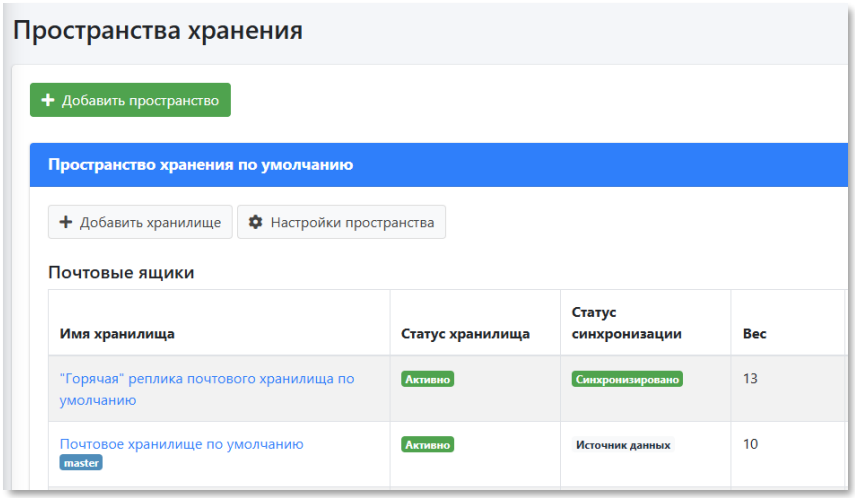
- выбрать раздел «Настройки», пункт «Пространства хранения», в открывшемся окне нажать кнопку «Добавить хранилище» и создать хранилище с типом «Почтовые ящики», ролью slave и весом большим, чем имеющееся основное хранилище по умолчанию, в разделе «Подключения» указать

параметры хранилища, планируемого как основное, для протокола NFS

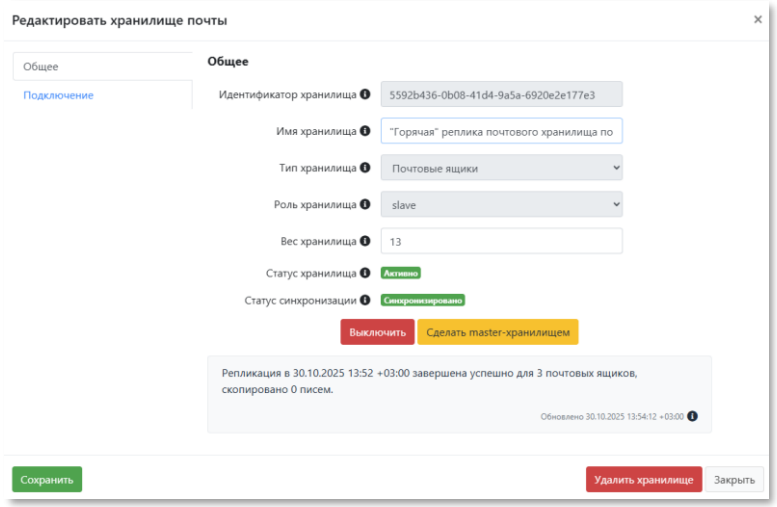




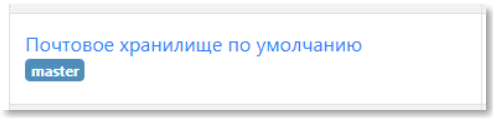
- дождаться создания и синхронизации нового хранилища



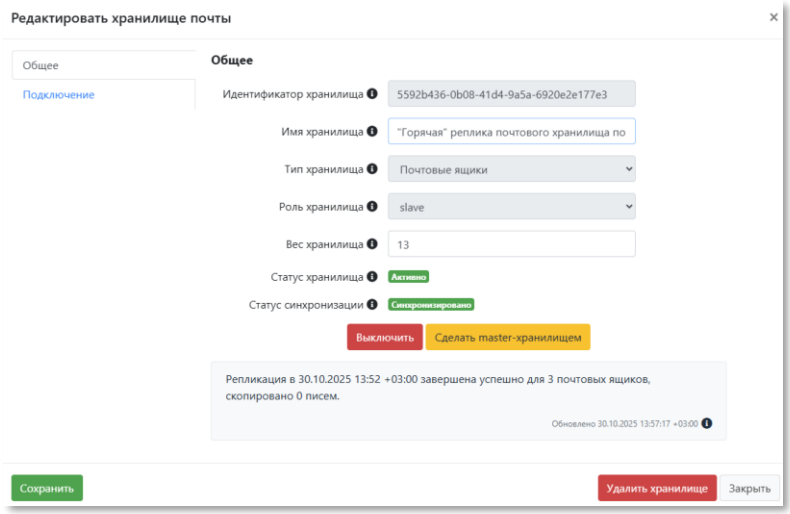
- открыть окно свойств вновь созданного NFS хранилища в списке и нажать кнопку «Сделать мастером»



- убедиться, что роль хранилища изменилась со slave на master, при этом роль локального хранилища сменится с master на slave



- выбрать в списке локальное хранилище, отрыть его и выбрать последовательно пункты «Выключить», «Сохранить», «Удалить хранилище», «Сохранить»



- убедиться, что локальное хранилище удалено

Внимание!

При выполнении удаления хранилища из графического интерфейса администратора RuPost происходит только размонтирование точки хранения, **данные почтовых ящиков и структура каталогов MailDir не удаляются**.

Для полного удаления данных необходимо вручную удалить их в соответствующем каталоге точки монтирования NFS.

2.8.2. Рекомендации по настройке сетевого файлового хранилища на примере NFS сервера в составе ОС Astra Linux

В примере рассматривается конфигурация, рассчитанная для обслуживания до 20000 почтовых ящиков, при среднестатистической ежедневной нагрузке 60% на чтение и 40% на запись. При указанной нагрузке к системным требованиям к серверу относятся:

- 4 ядра CPU;
- 4 GB оперативной памяти;

При этой конфигурации:

- для версии Astra 1.7.4 и ниже в файле /etc/default/nfs-kernel-server рекомендуется выставить параметр RPCNFSDCOUNT, отвечающий за количество обработчиков соединений, равным произведению 32 на количество процессорных ядер, т.е. для текущего примера:

```
# Number of servers to start up
RPCNFSDCOUNT=128
```

- для версии ОС, начиная с Astra 1.7.5 и выше, в файле /etc/nfs.conf раскомментировать строку threads в блоке [nfsd] и выставить значение

```
[nfsd]
threads=128
```

Также для приведённых далее параметров ядра Linux следует выставить значения буферов в соотношении 1 MiB на каждый 1 GB оперативной памяти:

```
net.core.wmem_max = 4194304
net.core.rmem_max = 4194304
net.core.wmem_default = 4194304
net.core.rmem_default = 4194304
```

При более чем четырёх узлах в кластере RuPost рекомендуется выделить для сервера NFS одно процессорное ядро и 1 GB оперативной памяти для обслуживания каждого дополнительно узла; таким образом для шести узлов RuPost требуется 6 CPU/6GB RAM.

В связи с тем, что задержки и производительность NFS связанны с индивидуальной нагрузкой и активностями той или иной организации, рекомендуется вносить коррективы в настройки и используемое оборудование при столкновении с недостаточной производительностью NFS сервера, а также вести мониторинг следующих параметров:

- Производительность дисковой системы ввода/вывода такими инструментами как iostat. При 100% утилизации мощности системы хранения данных принимать шаги по увеличению IOPS.
- Загрузка процессоров. При чрезмерной загрузке увеличивать количество ядер, объём доступной памяти, вносить изменения в системные настройки согласно рекомендациям, описанным выше.
- Утилизация пропускной способности сетевых соединений. В том числе, задействуя инструменты nfsstat на предмет увеличивающегося количества ретрансмит-пакетов. При обнаружении указанной проблемы увеличивать полосу пропускания сетевого трафика и вносить коррективы в MTU и настройки Jumbo Frames, если это возможно для топологии сетевых коммуникаций.

2.8.3. Рекомендации по настройке количества inode для хранилищ почтовых ящиков

Внимание!

Данная настройка рекомендуется для MailStorage, MailArchive и MailRecord.

В RuPost для хранения почты используется Maildir — формат хранения почты, при котором каждое письмо сохраняется как отдельный файл в файловой системе. Это означает, что при большом количестве писем критически важным ресурсом становится не только объём диска, но и количество inode — индексов файлов, необходимых для хранения метаданных каждого файла. Это справедливо для всех типов хранилищ: почты, архивов и RecordStore

Если inode заканчиваются, система не сможет создать новые файлы, даже если доступно свободное дисковое пространство.

Для Maildir-хранилища предпочтительны следующие файловые системы:

- **ext4** — требует настройки плотности inode вручную при создании.
- **BTRFS**, **XFS**, **ZFS** — динамически выделяет inode по мере необходимости.

В случае, когда используется файловая система ext4, по умолчанию создаётся один inode на 16 КБ пространства. Для Maildir этого может быть недостаточно. Рекомендуется использовать следующую команду для увеличения числа inode:

```
mkfs.ext4 -i 4096 /dev/sdX1
```

- Ключ -i 4096 указывает системе создавать один inode на каждые 4 КБ данных.
- Это увеличивает общее количество inode в четыре раза по сравнению со стандартной настройкой.
- Подходит для систем, где предполагается большое количество мелких файлов.

Для проверки текущего количества inode:

```
df -i или tune2fs -l /dev/sdX1
```

Количество inode в файловой системе ext4 напрямую зависит от размера раздела и параметра i (bytes-per-inode), задающего, на какое количество байт данных создаётся один inode.

```
Количество inode = Общий размер раздела (в байтах) / i
```

Примеры расчета количества inode:

Размер раздела	I (байт на inode)	Кол-во inode
100 ГБ	16384	~6 553 600
100 ГБ	4096	~26 214 400
500 ГБ	4096	~131 072 000

1 ТБ	4096	~262 144 000

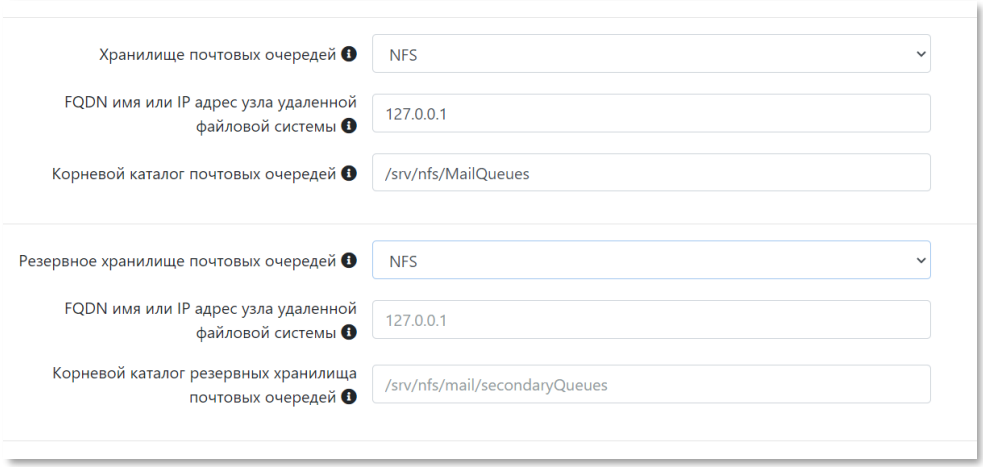
Слишком большое количество inode не опасно, но может иметь негативные побочные эффекты (особенно если это число сильно завышено по сравнению с реальной потребностью) – а именно:

- Таблица inode занимает дисковое пространство. Часть диска будет зарезервирована под inode, и, при небольшом числе файлов, она никогда не будет использоваться.
- Дольше выполняется проверка и монтирование файловой системы.
- Увеличивает объем метаданных, что может замедлить операции с файловой системой.

2.9. Резервный NFS для очередей Postfix

В версии 3.1.0 для повышения отказоустойчивости RuPost добавлена возможность использования резервного NFS сервера для очередей Postfix. Если на одном из узлов кластера подключение к основному NFS, на котором расположены очереди Posfix, выйдет из строя, то данный экземпляр RuPost автоматически переключится на работу с резервным NFS сервером без остановки обработки почты. При этом, очереди Postfix переключившегося экземпляра, оставшиеся на основном NFS, будут эвакуированы другими экземплярами RuPost.

Настроить резервный NFS сервер для очередей Postfix можно в Панели управления (“Общие настройки” -> “Почта”).



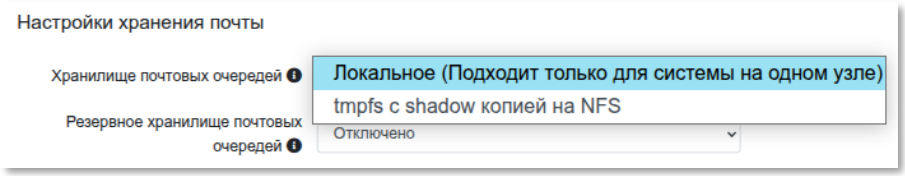
В версии 4.2.0 оптимизирована работа компонента *rupost-mta* (Postfix).

Почтовые очереди *incoming*, *active* и *deferred* перенесены в оперативную память, что существенно ускорило обработку почты и снизило нагрузку на дисковую подсистему или NFS.

Для того, чтобы при выходе из строя узла кластера исключить потерю почты, находящейся в почтовых очередях, резервная (shadow) копия сохраняется на NFS в каталоге *shadow*, который находится там же, где и

находились почтовые очереди до версии 4.2.0. С учетом этого изменения, были, также, внесены изменения в механизм эвакуации почты при выходе узла кластера из строя.

Для одноузловой конфигурации, где для хранения почты используется локальный диск, перенос очередей в оперативную память не применяется.



Внимание!

Скорость работы системы RuPost принципиально зависит от скорости доступа к ресурсам NFS в части сетевого взаимодействия и скорости выполнения операций чтения-записи. Для обеспечения быстрого отклика системы крайне рекомендуется использовать высокоскоростные диски **SSD** для оперативного хранилища почты **MailStorage**, shadow копии почтовых очередей и индексных файлов **IndexFiles**.

Внимание!

Для корректной работы, необходимо убедиться, что на сервере NFS и всех клиентах NFS время синхронизировано.

Если внутренние часы узлов отличаются друг от друга более чем на одну секунду и несколько клиентов одновременно обращаются к одному и тому же почтовому ящику, в работе сервисов могут появляться критические ошибки.

Внимание!

В случае использования NFS на базе Astra Linux необходимо использовать ALSE версии не ниже 1.7.4.

2.10. Настройки DNS

Для обмена письмами в сети интернет необходимо зарегистрировать RuPost на серверах DNS.

В корпоративном DNS для отправки и получения писем с внутрикорпоративных клиентов и SMTP серверов требуется настроить:

- **A запись** для имени хоста почтовой системы (заполняется в общих настройках под заголовком «Имя узла почтовой системы в DNS MX записи»), указывающую на локальный IP адрес узла RuPost или в кластерной конфигурации — внутренние адресы балансиров почтовой системы. В случае использования конфигурации с релей-сервером, за которым находятся серверы RuPost, в этой записи указывается IP адрес и имя хоста релей-сервера.

- **CNAME к A записи,** указанной выше. В случае указания конкретного сервера, можно задать приоритет обращения через Weight. Запись может состоять из нескольких адресов.

- Наличие **PTR записи** внутри сети зависит от Ваших корпоративных политик для других SMTP серверов, сосуществующих в организации. Например, используемых как open-relay для принтеров/МФУ и других систем.

- **Тип TXT (SPF)** со значением: v=spf1 a -all (опционально, внутри обычно не используется).

Возможно использование Split-DNS, если данный DNS сервер публично является authoritative для почтового домена, либо корректно настроена DNS-пересылка. Уточняйте возможность у Вашего провайдера услуги.

Для общедоступного (публичного) DNS:

- **A запись** для имени хоста почтовой системы (заполняется в общих настройках под заголовком «Имя узла почтовой системы в DNS MX записи»), указывающую на IP адресы шлюзов или балансиров почтовой системы, если они имеют публичные IP адресы. В случае использования конфигурации с релей-сервером, за которым находятся серверы RuPost, в этой записи указывается IP адрес и имя хоста релей-сервера.

- **PTR запись** для публичного IP адреса шлюза, с адреса которого набор узлов RuPost выполняет отправку сообщений. Должна указывать на имя хоста, соответствующее вашему SMTP EHLO (указывается в общих настройках под заголовком «Имя узла почтовой системы в DNS MX записи»). В случае применения конфигурации с релей-сервером, через который серверы RuPost выполняют отправку сообщений, в этой записи указывается SMTP EHLO релей-сервера.

Для **каждого** почтового домена в обоих вариантах DNS ожидаются нижеуказанные записи.

- Тип **MX**, в которой указывается приоритет (вес) и имя хоста почтовой системы или релей сервера. Для всех обслуживаемых доменов упомянутое имя одинаково. Пример:

```
domain.ru. MX 10 mail.domain.ru.
```

- Тип **TXT** (SPF). В случае, когда публичный IP адрес шлюза, с адреса которого набор узлов RuPost выполняет отправку сообщений, *совпадает* с IP адресом **A записи** для имени хоста почтовой системы, значение будет следующим:

  v=spf1 mx ~all

- Однако, если набор узлов RuPost выполняет отправку сообщений с других публичных IP адресов, все они должны быть указаны в директиве **ip4**. Например, если отправка писем из системы RuPost может осуществляться от IP адреса **A** записи, указанной в **MX** записи почтового домена, а также от IP адресов *10.20.30.41*, *10.20.30.42*, то **SPF** запись будет иметь следующее значение:

  *v=spf1 mx ip4:*10.20.30.41 *ip4:*10.20.30.42 *~all*

- В случае применения конфигурации с релей-сервером, в **SPF** должны быть указаны IP адресы релей-серверов.

- Для каждого домена организации, с которых НЕ планируется рассылка писем, следует создать SPF или TXT запись со значением: ***v=spf1 -all***. В этом случае письма злоумышленников, пытающихся отправить от таких необслуживаемых почтовым сервером доменов, будут идентифицированы как нарушающие доверие большинством корректно настроенных почтовых систем получателей.

- Для автонастройки клиентских приложений (Evolution, Thunderbird и построенные на них клиенты) тип **CNAME** для домена следующего уровня по формуле:
  autoconfig.<почтовый-домен>.
  указывающая на имя хоста почтовой системы. Пример:
  ```
  autoconfig.domain.ru. CNAME mail.domain.ru.
  ```

- Для автонастройки клиента Outlook с плагином RuPost тип **CNAME** для домена следующего уровня по формуле:
  autodiscover.<почтовый-домен>.
  указывающая на имя хоста почтовой системы. Пример:
  ```
  autodiscover.domain.ru. CNAME mail.domain.ru.
  ```

- Для доступа клиентских приложений к контактам и корпоративным адресным книгам тип **SRV** для домена следующего уровня по формуле:
  _carddavs._tcp.<почтовый-домен>.
  указывающая вес, приоритет, 443 порт и имя хоста почтовой системы. Пример:
  ```
  _carddavs._tcp.domain.ru. SRV 0 1 443 mail.domain.ru.
  ```

- Для доступа клиентских приложений к календарям и задачам тип **SRV** для домена следующего уровня по формуле:
  _caldavs._tcp.<почтовый-домен>.
  указывающая вес, приоритет, 443 порт и имя хоста почтовой системы. Пример:
  ```
  _caldavs._tcp.domain.ru. SRV 0 1 443 mail.domain.ru.
  ```

Внимание!

В случае кластерной конфигурации, для корректной обработки **PTR** записей почтовых серверов, от которых RuPost получает письма, балансиры почтовой системы должны передавать соответствующие заголовки узлам RuPost, используя **Proxy Protocol**. Для работы протокола, необходимо в общих настройках почтовой системы (страница “*Общие настройки*” -> вкладка “*Кластер*”), указать в поле «*IP адреса внешних proxy*» все IP адреса балансиров почтовой системы, использующих **Proxy Protocol**.

3. Установка RuPost

Внимание!

Перед обновлением версии RuPost выполните **резервное копирование** узлов кластера и баз данных.

Внимание!

Если у вас в организации используются почтовые клиенты Desktop X и Workspad X рекомендуем, после завершения обновления RuPost, обновить сервер Workspad на актуальную версию.

Внимание!

Перед обновлением версии RuPost обратите внимание на выбор варианта обновления:

- **обычное** – все узлы кластера обновляются одновременно, но требуется предварительный вывод из эксплуатации всех экземпляров RuPost;

- **непрерывное** – обновление без прерывания обслуживания пользователей с последовательным обновлением узлов кластера.

Внимание!

После завершения **непрерывного** обновления возможно неравномерное распределение пользователей по узлам кластера.

При необходимости, для перераспределения пользователей на менее загруженные узлы, на всех узлах с большим количеством пользователей выполните команду:

```
rupost kick-local-users
```

Внимание!

В версии 3.0 изменена структура хранения почты. Перед обновлением, обязательно выполните резервное копирование хранилища почты (Maildir) и баз данных.

Внимание!

Перед началом установки RuPost должны быть подключены и доступны «base» и «extended» репозитории AstraLinux. При возникновении вопросов, связанных с обновлением операционной системы, обращайтесь в техподдержку ГК Астра.

Внимание!

После обновления с предыдущей версии необходимо повторно развернуть активную или выбрать новую конфигурацию на основании обновленных шаблонов конфигураций, устанавливаемых при обновлении системы.

Внимание!

После завершения установки версии 4.1.1 и выше, для обеспечения корректного ведения лога почтовых компонентов, необходимо выполнить команду:

```
sudo /usr/sbin/syslog-ng-ctl reload
```

Рекомендуется

Обновить операционную систему на узлах RuPost и NFS до версии AstraLinux 1.8.5 и более.

Рекомендуется

Установщик RuPost является терминальным. Для корректного отображения визуальной части рекомендуется к использованию следующие эмуляторы терминала:
- Терминал fly (fly-term)
- Терминал GNOME

3.1. Установка системы с помощью мастера установки

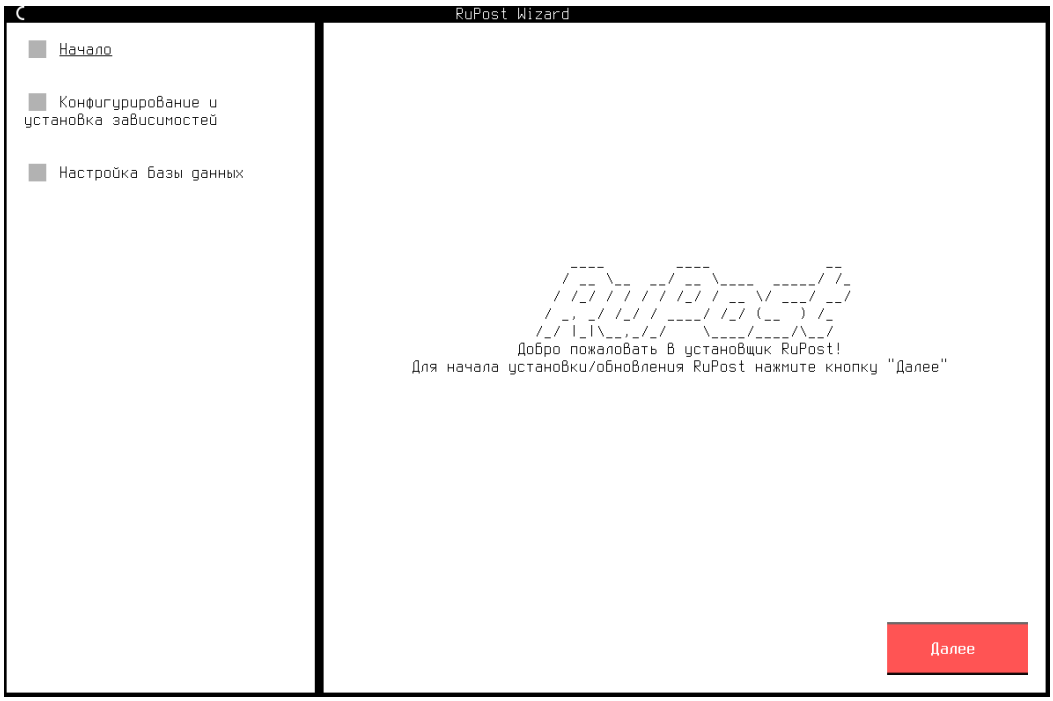
Внимание!

Начиная с версии 2.6.0 мастер установки позволяет установить либо обновить RuPost сразу на всех узлах кластера.

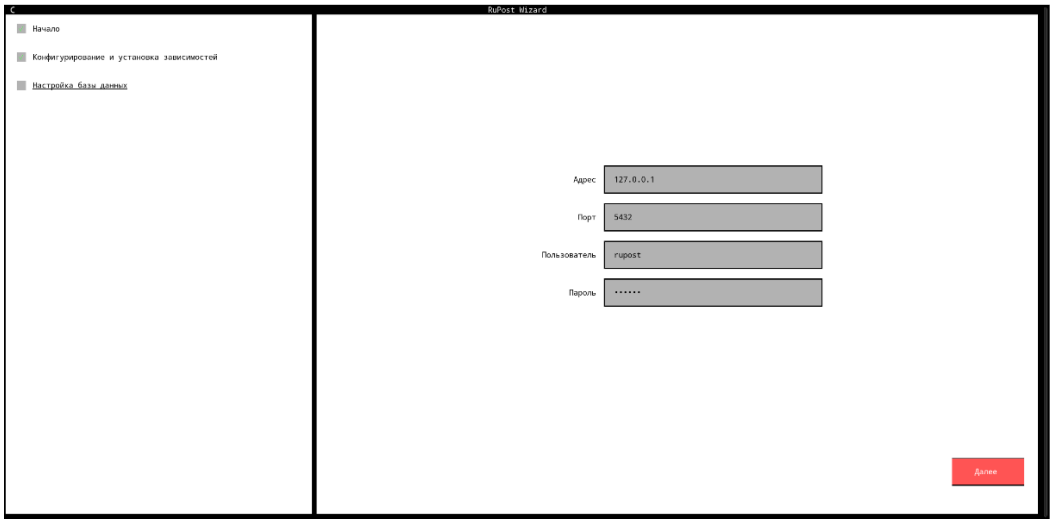
Для установки необходимо выполнить следующую команду:

```
sudo sh <путь к run-пакету>
```

запустится мастер установки RuPost:



На шаге создания баз данных необходимо указать информацию для подключения к СУБД:



Внимание!

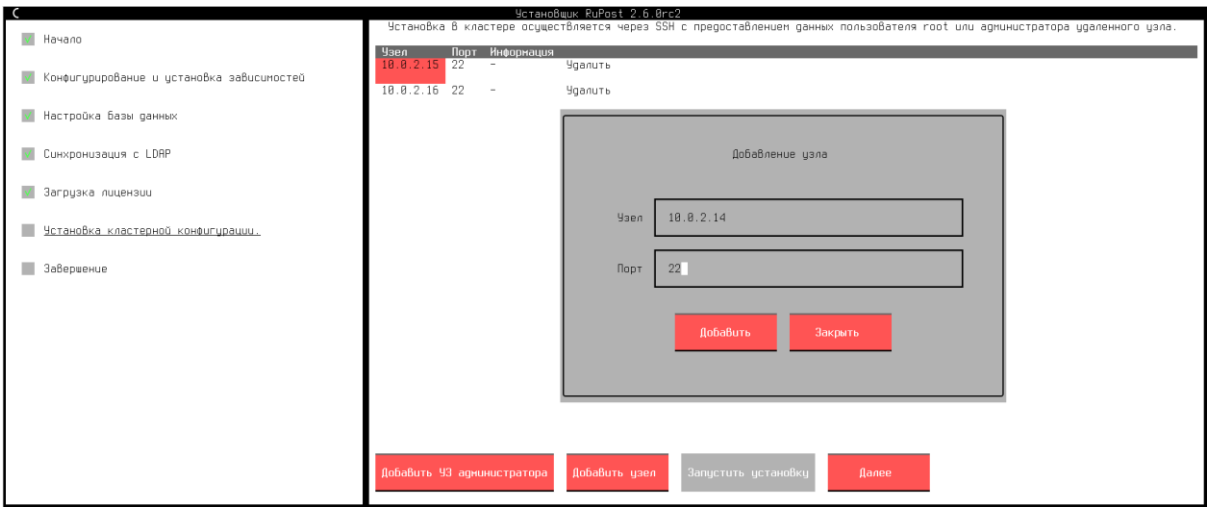
Программа RuPost при установке заводит сервисную учетную запись `rupost` для работы приложения. Запрещено самостоятельно создавать пользователя `rupost` администраторам системы.

При наличии лицензии редакции Enterprise можно выполнить установку или обновление кластера RuPost сразу на всех узлах.

Внимание!

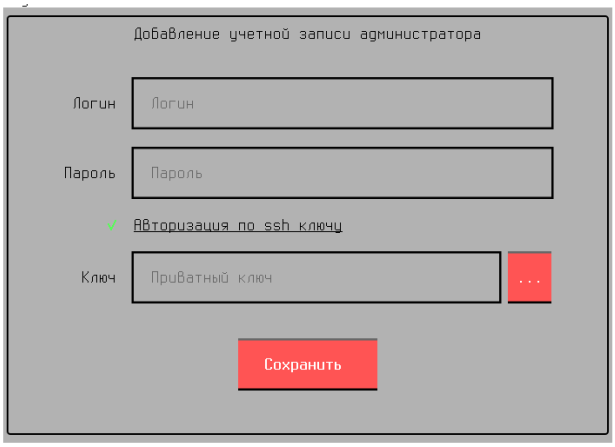
При подготовке к сетевой установке кластера RuPost, необходимо при установке операционной системы Astra Linux SE указать опцию “Доступ по SSH”.

Для установки системы в кластерной конфигурации нужно предварительно подготовить все узлы будущего кластера, установив на них операционную систему и задав реквизиты для входа – одинаковые на всех узлах.



Затем, используя кнопку “Добавить узел”, необходимо для каждого узла указать его IP адрес и SSH порт.

По кнопке “Добавить УЗ администратора” указываем реквизиты для подключения к узлам кластера:



После того, как информация об узлах и подключении к ним будет добавлена, по кнопке “Запустить установку” будет проведена установка RuPost на все указанные узлы и, затем, все экземпляры RuPost будут добавлены в кластер. Статус установки по каждому узлу будет отображаться в списке узлов в столбце “Информация”.

В процессе установки производятся следующие действия, результат которых записывается в лог файл `/var/log/rupost/monitor.log`:

1. Установка deb пакета.
2. Остановка сервисов:
    - nginx
    - sogo
    - dovecot
    - haproxy
    - postfix
3. Конфигурация параметров каталога глобальных sieve сценариев
4. Создание групп.
5. Создание пользователей.
6. Создание файлов.
7. Преднастройка установщика Postfix.
8. Обновление списка пакетов.
9. Установка зависимостей RuPost.
10. Проверка факта установки зависимостей.
11. Проверка версий установленных зависимостей.
12. Отключение автозапуска для компонентов.
13. Сброс статуса failed у компонентов.
14. Редактирование конфигурационного файла ldap.conf:
    - Доверять любым сертификатом LDAP без проверки.

- Не предлагать клиентам работать по рефералам. Критически важно для аутентификации в MSAD.

15. Настройка подсистемы Inotify
16. Отключение конфига nginx по умолчанию.
17. Удаление неподходящих инструментов, установленных в систему по умолчанию.
18. Настройка синхронизации времени.

Действия по настройке базы данных, сертификатов и таймеров:

1. Проверка наличия конфигурационного файла.
2. Ввод информации о базе данных при отсутствии конфигурационного файла или неудачной установке соединения с БД.
3. Запись конфигурационного файла при вводе информации о БД.
4. Проверка соединения с базой данных.
5. Возврат на шаг 2 при неудачной установке соединения с БД.
6. Отключение подписок служебного пользователя gal.
7. Применение миграций.
8. Запись версий.
9. Добавление шаблонов конфигураций почтовых компонентов.
10. Настройка сертификатов:
    - Генерируется уникальный самоподписанный SSL сертификат RuPost (rupost-builtin), необходимый для подключения клиентских приложений по SSL/TLS (например, Thunderbird).
    - Генерируется уникальный самоподписанный SSL сертификат (rupost-control-panel-builtin) для доступа к Панели управления RuPost по https.
    - Генерируется секретный ключ для протокола Диффи-Хеллмана (файл 'rupost-builtin-dhparam.pem', автоматически загружаемый в конфигурационную базу данных системы).
11. Включение наблюдателя за конфигурационными файлами.
12. Перезапуск rupost.
13. Перезапуск наблюдателя за конфигурационными файлами.
14. Включение таймера mailqueue-evacuator.
15. Включение таймера distribution-lists-updater.
16. Настройка прав для файла конфигурации.

Действия по синхронизации с LDAP, добавлению лицензии и установки на удаленных узлах:

1. Синхронизация почтовых ящиков со службами каталогов LDAP.
2. Синхронизация фильтров LDAP со службами каталогов LDAP
3. Синхронизация динамических списков рассылки со службами каталогов LDAP.
4. Загрузка и проверка лицензии.
5. Установка на удаленных узлах при помощи ssh (при наличии лицензии редакции Enterprise).

Так как синхронизация с LDAP каталогами может занять существенное время, то синхронизацию можно выполнить и после завершения установки, выполнив команду CLI:

```
sudo rupost ldap sync
```

Внимание!

На ОС Astra Linux Special Edition с уровнем защищенности Воронеж или Смоленск необходимо после завершения установки RuPost задать уровень конфиденциальности для пользователя rupost:

sudo pdpl-user rupost -l 0:0

3.2. Командный интерфейс мастера установки (CLI)

В мастере установки поддерживается командный интерфейс CLI. Командный интерфейс предназначен для решения задач автоматизации развертывания экземпляров RuPost (настройка подключения к СУБД, определение администратора по-умолчанию и т.п.), управление которыми в дальнейшем осуществляется через Панель управления в браузере.

Внимание!

Для передачи аргументов командной строки при установке, нужно обязательно добавлять два тире [--] после `rupost-4.2.0-alse-amd64.run` и перед любыми передаваемыми аргументами.

```
Флаги вызова:
--help      вывести данное сообщение и выйти
```

`--silent` провести автоконфигурацию в "тихом" режиме, без использования графического конфигуратора

`--fast` флаг для быстрой установки.

Аргументы, используемые при флаге --silent:

`-h, --db-host`	`HOST`	HOST = адрес для подключения к СУБД
`-p, --db-port`	`PORT`	PORT = порт для подключения к СУБД
`-n, --db-name`	`NAME`	NAME = имя базы данных RuPost (rupost)
`-d, --data-db-name`	`NAME`	NAME = имя базы пользовательских данных (rupost_data)
`-sd, --shared-db-name`	`NAME`	NAME = имя общей базы данных
`-u, --db-user`	`USERNAME`	NAME = имя пользователя для подключения к СУБД
`--db-password`	`PASSWORD`	PASSWORD = пароль для подключения к СУБД (может быть передан и через переменную окружения `UPOST_INSTALLER_DB_PASSWORD`). Пароль должен быть заключен в одинарные кавычки.
`--patroni-cluster-host`	`HOST`	HOST = адрес узла patroni для подключения к СУБД
`--patroni-cluster-port`	`PORT`	PORT = порт узла patroni для подключения к СУБД

`-- skip-cluster-check` флаг пропуска проверки работающих экземпляров в кластере

`--sync-data-ldap` флаг полной синхронизации с LDAP. Для версии RuPost до 2.5.0, по умолчанию выполняется частичная синхронизация

`--set-timesync` флаг для выполнения настройки синхронизации времени

В версии 2.7.0 добавлена возможность ускоренной установки RuPost при обновлении с помощью опции `----fast` (флаг для “тихой” установки). При использовании этой опции, в ходе установки не производится обновление фильтров LDAP и динамических списков рассылки, что существенно сокращает время установки RuPost в случае больших каталогов LDAP.

Пример вызова:

```
sudo sh rupost-4.2.0-alse-amd64.run -- --silent --db-host 127.0.0.1 --db-port 5432 --db-user rupost --db-password "12345678" --db-name rupost --data-db-name rupost_data --skip-cluster-check
```

Пример вызова (расположение СУБД PostgreSQL в кластере patroni):

```
sudo sh rupost-4.2.0-alse-amd64.run -- --silent --patroni-cluster-host 192.168.186.64 --patroni-cluster-port 8008 --db-user rupost --db-password "12345678" --db-name rupost --data-db-name rupost_data --skip-cluster-check
```

4. Обновление системы

Версия RuPost 4.0 поддерживает ОС:

- Astra Linux Special Edition (ALSE) 1.7.x (1.7.4 и выше)
- Astra Linux Special Edition (ALSE) 1.8.x (1.8.1, 1.8.2, 1.8.4 и выше)

Внимание!

В случае развертывания на версии AstraLinux 1.8.3 необходимо обратиться в службу поддержки для получения инструкций по использованию и настройке корректной версии pg_pool.

Для Astra Linux Special Edition (ALSE) 1.7 версия ядра должна быть linux-5.15-generic и выше.
Для Astra Linux Special Edition (ALSE) 1.8 версия ядра должна быть linux-6.1-generic и выше.

Для соответствующих основных версий Astra Linux необходимо использовать предназначенные для них дистрибутивы - установочные пакеты:

Операционная система	Установочный пакет RuPost
Astra Linux Special Edition 1.7.6 и выше	rupost-4.2.0-alse-amd64.run

RuPost поддерживает работу Astra Linux Special Edition в режиме защищенности «Воронеж» и «Смоленск» при выставленных по умолчанию параметров безопасности (Мандатный контроль целостности, Мандатное управление доступом, запрет трассировки ptrace, запрет пароля для команды sudo).

Перед установкой RuPost должен быть подключен расширенный репозиторий Astra Linux. При установке RuPost из данного репозитория будут установлены дополнительные пакеты:

```
lua-json lua-lpeg liblasso3 python3-tzlocal patch
```

Рекомендуется

Обновить операционную систему на узлах RuPost и NFS до версии AstraLinux 1.8.5.

Внимание!

Если у вас в организации используются почтовые клиенты Desktop X и Workspad X рекомендуем, после завершения обновления RuPost, обновить сервер Workspad на актуальную версию.
Если используется WorksPad сервер версии 7.0 и выше, то меняется формат его адреса – новый формат: https://<WorksPadGateway>/rupostapi

Внимание!

Перед обновлением версии RuPost обратите внимание на выбор варианта обновления:

- **обычное** – все узлы кластера обновляются одновременно, но требуется предварительный вывод из эксплуатации всех экземпляров RuPost;

- **непрерывное** – обновление без прерывания обслуживания пользователей с последовательным обновлением узлов кластера.

Внимание!
После завершения **непрерывного** обновления возможно неравномерное распределение пользователей по узлам кластера.

При необходимости, для перераспределения пользователей на менее загруженные узлы, на всех узлах с большим количеством пользователей выполните команду:

```
rupost kick-local-users
```

Внимание!
При обновлении системы с предыдущей версии обычным методом (с выводом из эксплуатации всех экземпляров) требуется:
1. Выполнить резервное копирование узла, на котором развернут(ы) RuPost
2. Вывести из эксплуатации все экземпляры RuPost
3. Остановить сервис rupost на всех узлах. Это можно сделать командой:
`sudo systemctl stop rupost`
3. Установить новую версию на всех узлы системы
4. Повторно развернуть активную конфигурацию с использованием обновленной версии необходимого шаблона конфигураций.

Рекомендуется
Для того, чтобы обновить пользовательские архивы в соответствии с новой схемой, примененной в версии 4.1.0, добавлена специальная команда CLI
`rupost mailbox update-archive-emails-folders`
Рекомендуется выполнить данную команду сразу же после установки версии 4.1.0.

После обновления системы нет необходимости осуществлять ввод в эксплуатацию экземпляров RuPost – они будут автоматически введены в эксплуатацию после обновления с сохранением всех конфигурационных параметров.

4.1. Непрерывное обновление кластера

Внимание!
При использовании **PostgreSQL версии 11 и ниже** - перед использованием непрерывного обновления версии 3.3.0 добавьте необходимые базы данных и расширения *ltree* и *pgcrypto*. Для этого выполните следующие команды (выполняются с правами суперпользователя):

```
sudo -u postgres psql -c 'create database rupost_gamma with encoding 'utf8' owner
rupost;'
sudo -u postgres psql -c 'create database rupost_logs_gamma with encoding 'utf8'
owner rupost;'
sudo -u postgres psql -d rupost -c 'create extension if not exists ltree;'
sudo -u postgres psql -d rupost -c 'create extension if not exists pgcrypto;'
sudo -u postgres psql -d rupost_gamma -c 'create extension if not exists ltree;'
sudo -u postgres psql -d rupost_gamma -c 'create extension if not exists pgcrypto;'
```

Если при установке RuPost было использовано имя базы данных, отличное от `rupost` то замените `rupost` на актуальное название базы данных в вышеприведенных командах.

Внимание!

Непрерывное обновление не поддерживает silent режим.

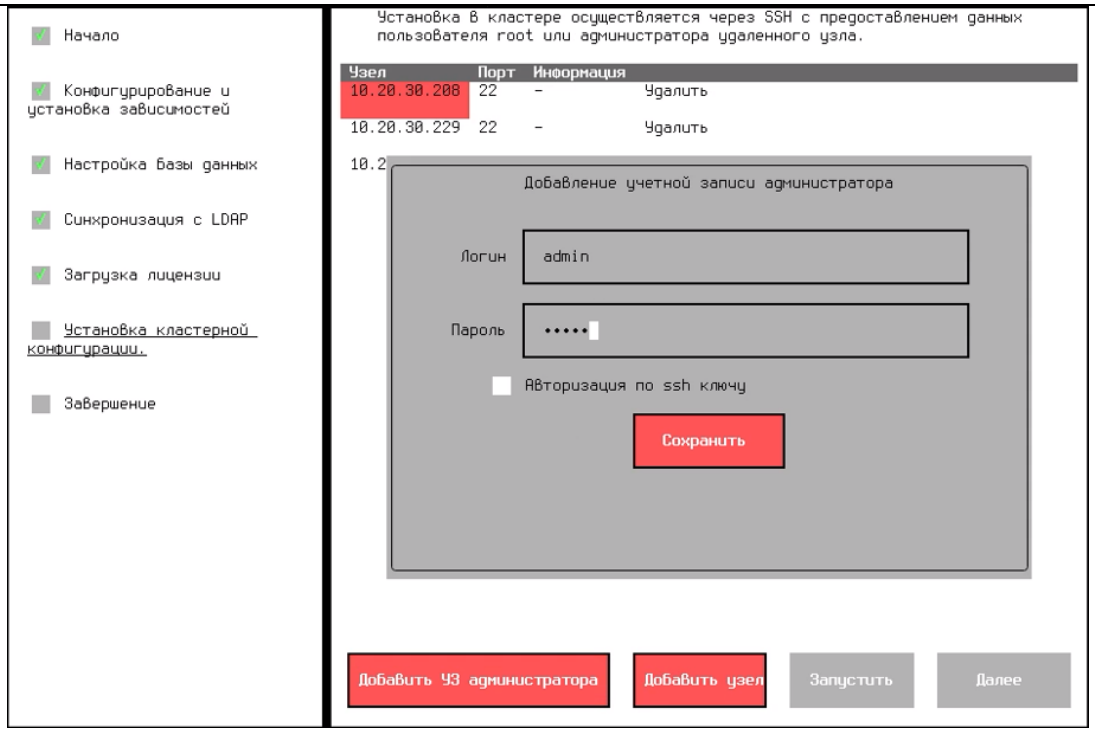
В версии 3.2.0 реализована возможность последовательного обновления узлов кластера на новую версию RuPost - “непрерывное обновление”:

При обновлении **на версию 3.2.0** – установка с минимальной остановкой обработки почты (обычно – менее минуты);

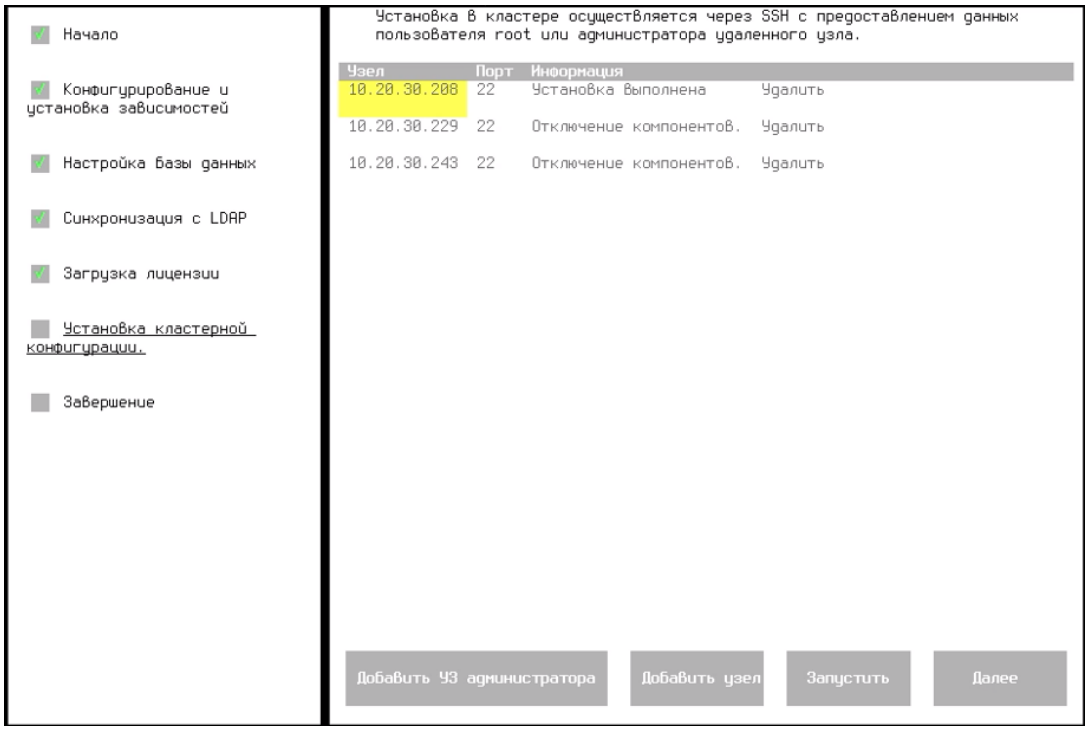
При обновлении **с версии 3.2.0** – будет происходить установка без остановки обработки почты.

Для использования этого режима обновления, необходимо на одном из узлов кластера запустить установку RuPost и на шаге выбора типа установки подтвердить выбор варианта “непрерывное обновление”.

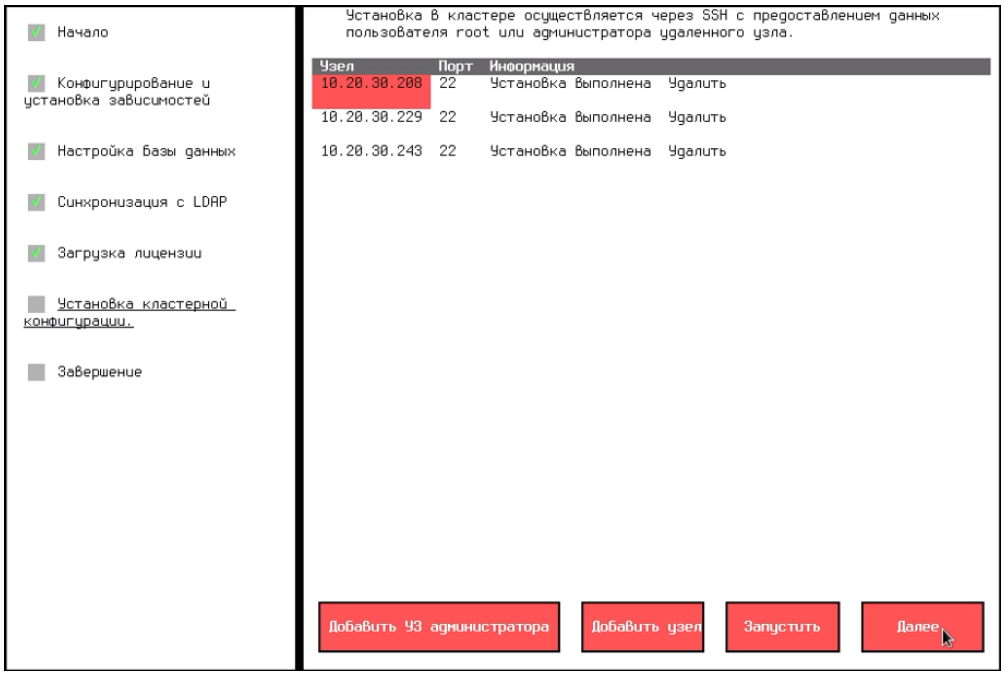
На странице списка узлов кластера необходимо нажать кнопку “Добавить УЗ администратора” и указать данные учетной записи, имеющей права администратора на каждом узле кластера.



После этого будет запущен процесс обновления узлов кластера – статусы узлов на каждом шаге обновления будут отображаться на странице установки в столбце “Информация”.

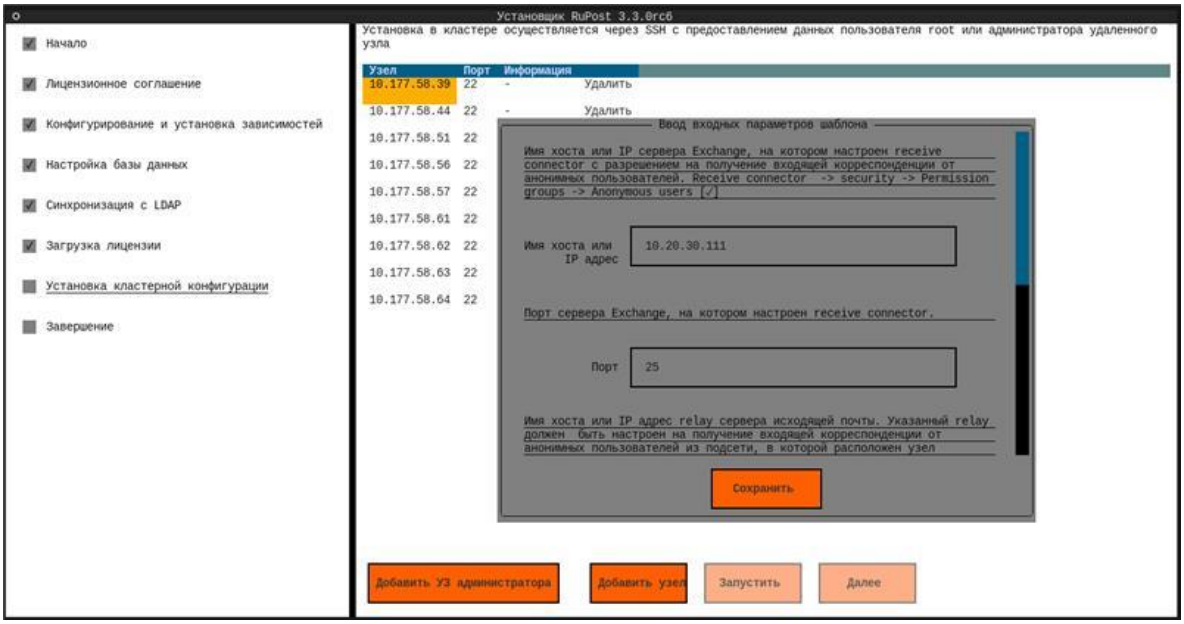


По окончанию установки каждый узел будет находиться в статусе “Установка выполнена”.



В версии 3.3.0 в процедуру установки RuPost в режиме “непрерывное обновление” добавлен шаг дополнительного контроля значений, задаваемых администратором, для вводимых параметров встроенных шаблонов.

Дополнительная проверка значений параметров необходима в том случае, когда в ходе установки RuPost проводится обновление встроенных шаблонов, содержащих вводимые параметры.



При обновлении кластера RuPost, проверка значений параметров происходит во время обновления первого узла кластера. В случае, если администратор изменил значения параметров на этом шаге установки, установка на всех узлах кластера будет проведена с новыми значениями параметров.

5. Подготовка системы к реальной эксплуатации (меры информационного контроля)

Внимание!

Совокупность представленных ниже мер требуется для снижения рисков.

5.1. Использование действительных корпоративных сертификатов

Внимание!

При создании ключей необходимо выполнить следующие требования:
- для алгоритма шифрования RSA, рекомендуемая длина ключа должна составлять не менее 2048 бит;
- для алгоритма шифрования на основе эллиптических кривых ECDSA - длина ключа не менее 256 бит;

Внимание!

Генерируемые самоподписанные сертификаты требуют для реальной эксплуатации замены на действительные (валидные) сертификаты с использованием корпоративного Удостоверяющего Центра (УЦ – CA, Certificate Authority).

Генерируемый сертификат должен включать в себя полную цепочку промежуточных сертификатов.

При продолжении использования самоподписанных сертификатов вы будете получать ошибки. Например, при входе в Панель администратора вы увидите предупреждение о вероятной угрозе.
Поддерживаемые типы сертификатов:

- mail - SSL сертификат почтового сервера RuPost
- control_panel - SSL сертификат панели управления

Для добавления собственных сертификатов, используемых при подключении клиентских приложений, необходимо выполнить команду:

```
rupost cert add "mail_cert" \
    --cert-type mail \
    --cert-path certs/rupost-mail.crt \
    --key-path certs/rupost-mail.key
```

где:

- `mail_cert` - имя сертификата которое будет отображаться в БД
- `mail` - тип сертификата
- `certs/rupost-mail.crt` - путь до сертификата
- `certs/rupost-mail.key` - путь до ключа

Добавление сертификата для Панели управления аналогично процессу добавления сертификата для клиентских приложений. Ключевым отличием является то, что используется другой тип:

```
rupost cert add "control_panel_cert" \
```

```
    --cert-type control_panel \
    --cert-path certs/rupost-control-panel.crt \
    --key-path certs/rupost-control-panel.key
```

Внимание!

Сертификат типа "control_panel" должен включать в себя полную цепочку промежуточных сертификатов.

При добавлении сертификата с типом "control_panel" он не будет применен до принудительной перезагрузки сервиса rupost. В случае кластера это необходимо сделать на всех узлах системы.

Обновление системы с использованием интерактивного конфигуратора останавливает все компоненты системы на выбранном узле. После обновления требуется переразвернуть конфигурацию почтовых служб RuPost с использованием обновленной версии необходимого шаблона конфигураций. При запуске конфигуратора сохраняются параметры системы - общие настройки, зарегистрированные службы каталогов, почтовые домены, почтовые ящики и т.п.

6. Действия после установки и настройки системы

Дальнейшая настройка и управление RuPost осуществляется через командный интерфейс или в графической **Панели управления** RuPost, описанным в *Руководстве администратора*.

Панель управления доступна из web-браузера по имени или адресу узла RuPost по порту 5000, например локально (находясь на выбранном узле):

`https://localhost:5000`

или по имени хоста почтовой системы:
`https://mail01.demo.local:5000`

Список доступных команд RuPost CLI можно получить, выполнив команду

`sudo rupost` или `sudo rupost --help`

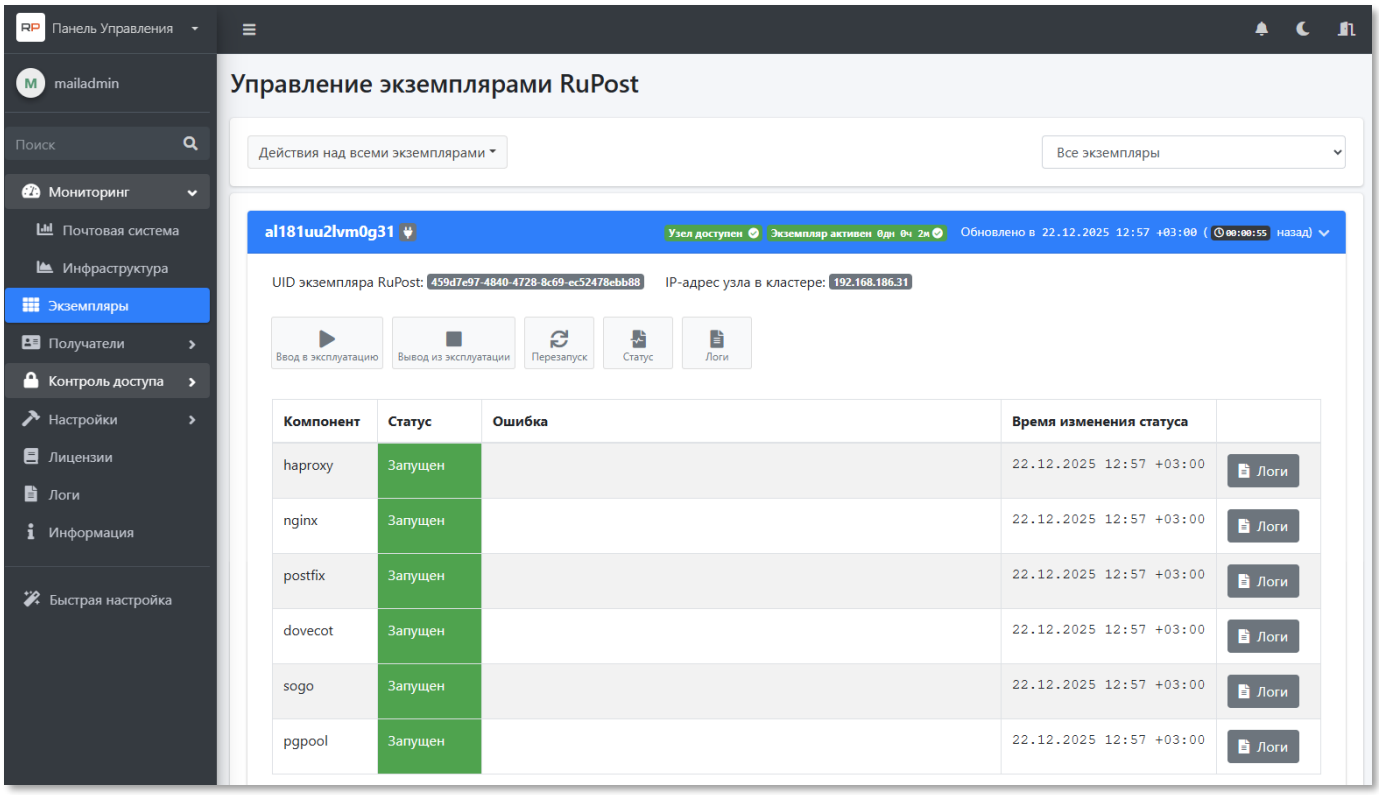
```
root@al174m0g06:~# rupost --help
 RuPost CLI

┌─ Параметры ──────────────────────────────────────────────────────────────────────────────────────────────────────────┐
│ --help      Выводит данную подсказку.                                                                                │
└──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┘
┌─ Команды ────────────────────────────────────────────────────────────────────────────────────────────────────────────┐
│ about                                                Вывести краткую сводку о приложении.                           │
│ admins                                               Группа команд для управления...                                 │
│ audit                                                Вывести лог активности администраторов.                         │
│ autostart                                            Группа команд для управления...                                 │
│ cert                                                 Группа команд для управления сертификатами.                     │
│ components                                           Группа команд для управления почтовыми...                       │
│ db                                                   Группа команд для управления базой данных.                      │
│ distribution-lists                                   Группа команд для управления почтовыми...                       │
│ impersonation                                        Группа команд для управления аккаунтом...                       │
│ instances                                            Группа команд для управления экземплярами...                    │
│ ldap                                                 Группа команд для взаимодействия с LDAP.                        │
│ ldap-filters                                         Группа команд для управления фильтрами LDAP.                    │
│ licenses                                             Группа команд для управления лицензиями.                        │
│ logs                                                 Вывести журнал всех почтовых компонентов.                       │
│ mailboxes                                            Группа команд для управления почтовыми...                       │
│ mailboxgroup                                         Группа команд для управления почтовыми...                       │
│ mailqueues                                           Группа команд для управления почтовыми...                       │
│ mailspace                                            Группа команд для управления...                                 │
│ mailstore                                            Группа команд для управления хранилищами...                     │
│ make-gal                                             Выполнить обновление корпоративной...                           │
│ push                                                 Группа команд для управления push...                            │
│ report                                               Собрать всю информацию о работе экземпляра...                   │
│ resources                                            Группа команд для управления ресурсами...                       │
│ restrictions                                         Группа команд для управления почтовыми...                       │
│ run                                                  Запустить приложение.                                           │
│ sieve                                                Группа команд для настройки работы с...                         │
│ template                                             Группа команд для управления шаблонами...                       │
│ user-audit                                           Группа команд для управления логированием...                    │
└──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┘
```

Внимание!
Часть функций RuPost доступна только через командный интерфейс CLI или Панель управления.

7. Удаление RuPost из операционной системы

Внимание!

Перед выполнением удаления экземпляра RuPost необходимо записать UID экземпляра RuPost уникальный `instance_id / UUID`, который можно получить на экране «Экземпляры»:



Для удаления RuPost выполните скрипт

```
/usr/bin/uninstall-rupost
```

Внимание!

Почтовые ящики, размещенные в файловой системе, не удаляются. Файл лицензии не удаляется. Файл конфигурации config.json экземпляра (узла) RuPost удаляется (см. *“Руководство администратора RuPost”*).

Внимание!

При удалении скриптом:

```
/usr/bin/uninstall-rupost
```

экземпляр в кластерной конфигурации, на котором была исполнена команда, не пропадает из списка Экземпляров на остальных рабочих экземплярах. Для удаления данного экземпляра из списка Экземпляров необходимо на любом из оставшихся рабочих экземпляров выполнить команду:

```
rupost instances delete <instance_id>
```

где instance_id уникальный идентификатор удаленного экземпляра (сохраненный перед удалением).

Пример команды:

```
rupost instances delete '01234567-89ab-cdef-0123-456789abcdef'
```

8. Основные пути и файлы системы

Ключевые файлы и каталоги, необходимые администратору системы:

Путь	Описание
/usr/bin/rupost	Главный исполняемый файл RuPost
/usr/bin/rupost-wizard	Интерактивный конфигуратор
/etc/rupost	Директория с файлом настроек config.json и файлом лицензии редакции Standard
/etc/systemd/system/rupost*.service **/etc/systemd/system/rupost*.timer**	Службы RuPost (unit-файлы): /etc/systemd/system/rupost.service /etc/systemd/system/rupost-distribution-lists-updater.service /etc/systemd/system/rupost-distribution-lists-updater.timer /etc/systemd/system/rupost-mailqueue-evacuator.service /etc/systemd/system/rupost-mailqueue-evacuator.timer /etc/systemd/system/rupost-scheduler.service
/var/log/rupost/monitor.log	Файл журнала RuPost
/usr/lib/rupost	Каталог с репозиторием, автогенерируемым самоподписанным сертификатом и вспомогательными программами и библиотеками, необходимыми для корректного функционирования RuPost

Системные журналы компонентов системы:

- Объединённый лог Rupost-mda (dovecot) и Rupost-mta (postfix) располагается по пути `/var/log/mail.log`. Он удобен для того, чтобы отслеживать факт отправки/получения писем и взаимодействие этих двух ключевых почтовых компонентов.
- Отдельно лог Rupost-mda (dovecot) можно вывести в stdout командой `journalctl -u dovecot` либо сохранить во временный файл `journalctl -u dovecot > /tmp/dovecot.log`.
- Лог встроенного web-клиента и сервера календарей и контактов почтового компонента Rupost-mua (SOGo) располагается по пути `/var/log/sogo/sogo.log`.

- Лог веб прокси-сервера Rupost-mp (Nginx) с информацией по доступу к ресурсам пишется в файл `/var/log/nginx/access.log`, а сообщения об ошибках в работе того же компонента можно получить в файле `/var/log/nginx/error.log`.
- Отдельно лог Rupost-lb (HAProxy) можно вывести в stdout командой `journalctl -u haproxy` либо сохранить во временный файл `journalctl -u haproxy > /tmp/ haproxy.log`.

Дополнительные файлы

Для корректной поддержки масштабируемости серверов RuPost система автоматически добавляет исключения для сервиса `dovecot.service` с целью снятия ограничений на число подключений к серверу. Для этого создается файл `/etc/systemd/system/dovecot.service.d/override.conf` и в нем автоматически устанавливаются следующие параметры сервиса:

- `LimitNOFILE=1048576`
- `LimitNPROC=4194304`

9. Средства диагностики

9.1. Единый сводный журнал (лог) для всех почтовых компонентов – команда CLI logs

```
root@al174m0g06:~# rupost logs --help
 Вывести журнал всех почтовых компонентов.

┌─ Аргументы ──────────────────────────────────────────────────────────────────
│ COMPONENTS    [[postfix|dovecot|nginx|sogo|haproxy|rupost|rupost-scheduler]]...
└──────────────────────────────────────────────────────────────────────────────
┌─ Параметры ──────────────────────────────────────────────────────────────────
│ --output  -o  ПАПКА  Путь, по которому будет сохранён лог. По умолчанию выводится в консоль.
│ --remote  -r         Получить журналы со всех узлов.
│ --help               Выводит данную подсказку.
└──────────────────────────────────────────────────────────────────────────────
```

В версии **2.5.0** добавлена возможность просмотра логов всех почтовых компонентов (в кластере – со всех почтовых компонентов текущего экземпляра RuPost) в виде единого лога, синхронизированного по времени событий. Таким образом, стало гораздо удобнее диагностировать работу сервисов RuPost в случае, когда, например, обработка почтового сообщения обеспечивается взаимодействием нескольких почтовых компонент.

Получить единый лог можно с помощью команды CLI:

```
sudo rupost logs
```

В версии **2.6.0** расширены возможности команды CLI `logs`:

1. Реализован сбор информации со всех экземпляров RuPost – в кластерной конфигурации достаточно подключиться только к одному экземпляру RuPost и получить логи всех почтовых компонентов со всех экземпляров.
   Для получения информации со всех экземпляров, выполните команду `logs` с параметром `-r`:

   ```
   sudo rupost logs -r
   ```

2. Добавлена опция components, позволяющая указать логи каких почтовых компонентов необходимо отображать.
   Например, для того, чтобы получить лог файл только для компонентов haproxy и postfix выполните команду:

   ```
   sudo rupost logs -components haproxy postfix
   ```

9.2. Поддержка сбора и экспорта логов – команда CLI report

В версии 2.5.0 добавлена возможность получить информацию о системе (в кластере – с одного узла), а также все логи работы почтовых компонентов в виде одного архива. Сформировать архив всех лог файлов можно с помощью команды CLI:

```
sudo rupost report
```

при этом формируется zip файл, содержащий следующий перечень файлов:

- app-report.txt
- hardware-report.txt
- postgres-report.txt
- licenses-report.txt
- monitor.log
- postfix.log
- dovecot.log
- nginx.log
- sogo.log
- haproxy.log

В версии 2.6.0 в команду CLI report добавлена возможность задавать диапазон времени для собираемых лог-файлов.

```
root@al174m0g06:~# rupost report --help
 Собрать всю информацию о работе экземпляра системы.

╭─ Параметры ──────────────────────────────────────────────────────────────
│ --output             -o    ПАПКА  Путь, по которому будет сохранён архив. По умолчанию /var/log/rupost.
│ --date-start         -ds   ТЕКСТ  Дата начала формирования аудита.
│ --date-end           -de   ТЕКСТ  Дата окончания формирования аудита.
│ --list-date-formats               Вывести поддерживаемые форматы ввода дат.
│ --help                            Выводит данную подсказку.
╰──────────────────────────────────────────────────────────────────────────
```

Перечень поддерживаемых форматов времени / дат можно получить командой:

```
sudo report –list-date-formats
```

```
root@al174m0g06:~# rupost report --list-date-formats
Поддерживаемые сокращения дат:
today             (04-07-24 16:42)
yesterday         (03-07-24 16:42)
week              (27-06-24 16:42)
month             (03-06-24 16:42)
year              (05-07-23 10:42)
Поддерживаемые форматы дат:
%H:%M             (16:42)
%d-%m             (04-07)
%d-%m %H:%M       (04-07 16:42)
%d-%m-%y %H:%M    (04-07-24 16:42)
%d-%m-%y          (04-07-24)
%m-%Y             (07-2024)
```

Например, для того, чтобы получить логи за сегодня, можно выполнить команду:

```
sudo report –ds today
```

9.3. Поддержка SOSReport

В версии 2.5.0 добавлена поддержка выгрузки системной информации и лог файлов через сервис SOSReport. Через этот сервис может быть получена та же информация, что и через команду CLI report:

```
sudo sos report -o rupost
```

При работе в кластере, для получения информации со всех узлов, используйте команду:

```
sudo sos collect -o rupost --nodes [список FQDN/IP адресов всех узлов кластера]
[параметры доступа к другим узлам]
```

10. Приложение 1. Расчёт системных требований в зависимости от планируемой нагрузки

10.1. Общие замечания к расчёту ожидаемых системных требований

Требования к оперативной памяти, числу ядер и производительности процессора зависят от числа обслуживаемых почтовых ящиков, а также общего количества пользователей, использующих WEB интерфейс для доступа к своим ящикам, календарям и контактам. В расчёт требуемой свободной оперативной памяти на каждом узле не включён объём, занимаемый операционной системой в полностью работоспособном состоянии. Поэтому к полученным для отдельного узла почтовой системы ожидаемым системным требованиям необходимо добавить занимаемые операционной системой ресурсы, а также зарезервировать 10% сверх полученного в расчётах результата. Также важным обстоятельством является то, что все узлы кластера почтовой системы должны обладать равными процессорами и объёмами оперативной памяти, а также иметь равные по производительности каналы связи.

Расчет ожидаемых ресурсов должен производиться для числа узлов RuPost, которые гарантированно останутся в эксплуатации в случае возникновения сбоя в ряде узлов кластера. То есть, если сбой произошел на половине узлов кластера, то оставшиеся в эксплуатации узлы должны обеспечить работоспособность всей почтовой системы.

При расчетах будет использоваться понятие `активных пользователей` — это число от общего числа заведенных пользователей в систему, которые в течении дня работают с почтовым сервером, заглядывают в календарь, просматривают/синхронизируют адресную книгу, читают и пишут почту. В это число включаются все пользователи если даже они один раз в день проверяют почту.

10.2. Оперативная память

Вне зависимости от активных пользователей, обслуживаемых почтовой системой, на каждом узле RuPost резервируется 2,6 ГБ. Для каждого активного пользователя, который в течении рабочего дня работает с почтовой системой получая доступ к своим ящикам, календарям и контактам, прибавляется по 45 МБ к указанному объёму.

В общих настройках почтовой системы RuPost на вкладке «Почта» добавлен раздел «Настройки веб-клиента». В нём требуется ввести значения для параметра «Число обработчиков WEB клиента». Он заполняется из расчёта по следующей формуле:

```
Число обработчиков WEB клиента = Число активных пользователей в течении дня / 5 (пять) / 1.6 (коэффициент использования обработчика) / число узлов почтовой системы
```

Однако параметр «Число обработчиков WEB клиента» имеется важный нюанс, его значение не допускается менее 20.

Обратите внимание, что если в логе `/var/log/sogo/sogo.log` будут появляться записи, содержащие строки «`No child available to handle incoming request!`», то это означает, что необходимо увеличить значение параметра «Число обработчиков WEB клиента» на некоторую величину, например, 10, и развернуть конфигурацию.

10.3. Процессор

Ожидаемые системные требования по процессору вычисляются по формуле:

```
На каждые 100 активных пользователей ожидается 1 ядро с округлением в большую
сторону.
```

Пример: для обслуживания 750 пользователей на двух узлах, минимальное число ядер на каждом узле будет вычислено следующим образом: 750 (общее число пользователей) / 100 (сто) / 2 (узла в кластере) = 4 ядра на каждом узле почтовой системы (округление в большую сторону).

Обратите внимание, что такой расчёт справедлив для процессоров уровня Intel Xeon E5-26xx v4 (Broadwell) или новее. Если используются процессоры более старых поколений, то к полученному числу необходимо применять поправочный коэффициент для увеличения числа вычислительных ядер, так как отсутствие современных программно-аппаратных решений в архитектуре процессора может значительно снижать производительность некоторых функциональных процессов, таких как терминация TLS соединений и других.

10.4. Дисковая память

Для установки всех компонентов RuPost, без учёта занимаемого операционной системой объёма в файловой системе, необходимо зарезервировать на каждом узле как минимум 20 ГБ в случае, когда почтовые ящики и очереди располагаются на сетевой файловой системе (NFS). Размер самого хранилища почтовых ящиков полностью зависит от интенсивности потока электронной корреспонденции, объёма передаваемых вложений и числа ящиков. Расчёт параметров хранилища выполняется из ожидаемого общего объёма почты без точных формул.

10.5. Подключения к базе данных PostgreSQL

Каждый узел кластера RuPost использует фиксированное число подключений, равное 55. Сверх этого каждый узел использует для каждого обработчика WEB клиента по три соединения. Общее число соединений можно получить по формуле:

```
Общее число соединений = (55 (фиксированное число) + число обработчиков WEB клиента
на одном узле * 3 + количество ядер на одном узле кластера * 10) * число узлов в
кластере RuPost
```

Для средней нагрузки число соединений составит 60-70% от расчетного числа

10.6. Пример расчёта минимальных системных требований

Предположим, что в организации необходимо обеспечить электронной почтой, календарями и корпоративными контактами 2000 пользователей. Для отказоустойчивости, система будет установлена на трёх узлах. В числе требований — ожидается регулярное единовременное обращение всех пользователей организации к почтовому WEB интерфейсу.

Количество активных пользователей порядка 95% это 1900 от 2000. Т.е. в течении дня к системе обращается 1900 пользователей.

Минимальная свободная оперативная память на каждом узле, без учёта потребления ресурсов операционной системой, должна составлять:

```
1900 (пользователи) / 3 (узлы) * 45 МБ / 1024 + 2,1 ГБ (константа) + 20% (на случай
отказа одного из узлов) = 37 ГБ
```

Необходимо убедиться, что в системе настроено использования swap и его размер не менее объема оперативной памяти, т. е. 37 ГБ.

На каждом узле применяется центральный процессор уровня Intel Core Processor (Haswell). Соответственно, число ядер на каждом узле RuPost минимально необходимо:

```
1900 (пользователи) / 100 (сто) / 3 (узлы) + 20% (на случай отказа одного из узлов) =
8 ядер (округление в большую сторону)
Определим число обработчиков WEB клиента. В общих настройках почтовой системы RuPost
на вкладке «Почта» в разделе «Настройки веб-клиента» параметру «Число обработчиков
WEB клиента» следует присвоить значение:
1900 (пользователи) / 5 (пять) / 1.6 / 3 (узлы) = 80 обработчиков (округление в
большую сторону)
```

Общее число подключений к базе данных PostgreSQL находится по указанной выше формуле:

```
(55 (фиксированное число) + 80 (число обработчиков WEB клиента на одном узле) * 3 + 8
(количество ядер на одном узле кластера) * 10) * 3 (узлы) = 1125 (общее число
подключений от кластера к базе данных)
```

Дисковое пространство – должны быть обеспечены свободные 40 ГБ на каждом узле. На сетевой файловой системе ожидается суммарно 5 ТБ почтовых данных (из расчета примерно 2,5 ГБ на пользователя).

11. Приложение 2. Функциональное взаимодействие RuPost с подключенными доменами LDAP

11.1. Права доступа к атрибутам у служебной учётной записи RuPost

11.1.1. FreeIPA

Служебная учётная запись должна обладать следующими правами на атрибуты пользователей RuPost.

Имя attributeTypes	Требуемые права
cn	чтение, поиск, сравнение
givenName	чтение, поиск, сравнение
l	чтение, поиск, сравнение
mail	чтение, запись, добавление, удаление, поиск, сравнение
mobile	чтение, поиск, сравнение
sn	чтение, поиск, сравнение
st	чтение, поиск, сравнение
street	чтение, поиск, сравнение
title	чтение, поиск, сравнение
employeeNumber	чтение, поиск, сравнение
employeeType	чтение, поиск, сравнение
facsimileTelephoneNumber	чтение, поиск, сравнение
ou	чтение, поиск, сравнение
pager	чтение, поиск, сравнение
proxyAddresses	чтение, запись, добавление, удаление, поиск, сравнение
objectClass	чтение, запись, поиск, сравнение
uid	чтение, поиск, сравнение
ipaUniqueID	чтение, поиск, сравнение

11.1.2. ALD Pro

Служебная учётная запись должна обладать следующими правами на атрибуты пользователей RuPost.

Имя attributeTypes	Требуемые права
cn	чтение, поиск, сравнение
givenName	чтение, поиск, сравнение

l	чтение, поиск, сравнение
mail	чтение, запись, добавление, удаление, поиск, сравнение
mobile	чтение, поиск, сравнение
sn	чтение, поиск, сравнение
st	чтение, поиск, сравнение
street	чтение, поиск, сравнение
title	чтение, поиск, сравнение
c	чтение, поиск, сравнение
employeeNumber	чтение, поиск, сравнение
proxyAddresses	чтение, запись, добавление, удаление, поиск, сравнение
rbtadp	чтение, поиск, сравнение
rbtamiddlename	чтение, поиск, сравнение
telephoneNumber	чтение, поиск, сравнение
objectClass	чтение, запись, поиск, сравнение
uid	чтение, поиск, сравнение
ipaUniqueID	чтение, поиск, сравнение

11.1.3. Active Directory

Служебная учётная запись должна обладать следующими правами на атрибуты пользователей RuPost.

Имя attributeTypes	**Требуемые права**
cn	чтение, поиск, сравнение
givenName	чтение, поиск, сравнение
l	чтение, поиск, сравнение
mail	чтение, запись, добавление, удаление, поиск, сравнение
mobile	чтение, поиск, сравнение
sn	чтение, поиск, сравнение
st	чтение, поиск, сравнение
street	чтение, поиск, сравнение
title	чтение, поиск, сравнение
company	чтение, поиск, сравнение
department	чтение, поиск, сравнение

facsimileTelephoneNumber	чтение, поиск, сравнение
homePhone	чтение, поиск, сравнение
pager	чтение, поиск, сравнение
proxyAddresses	чтение, запись, добавление, удаление, поиск, сравнение
objectClass	чтение, запись, поиск, сравнение
sAMAccountName	чтение, поиск, сравнение
objectGUID	чтение, поиск, сравнение

11.2. Функциональное использование объектных классов и атрибутов LDAP

11.2.1. Классы

В службах каталогов FreeIPA и ALD Pro (актуально для версии 1.1.1) все пользователи должны обладать объектным классом *ruPostMailAccount* (OID 1.3.6.1.4.1.57980.3.1.1.1). Данный класс выполняет две функции:

- позволяет учётным записям наследовать атрибут *proxyAddresses* (OID 1.3.6.1.4.1.57980.3.1.2.2);
- применяется в фильтрах привилегий и разрешений, которые накладываются на сервисную учётную запись RuPost в службе каталогов.

При расширении схемы, уже существующим в ldap пользователям не добавляется objectClass `ruPostMailAccount` - это можно сделать вручную командой:

```
ipa user-mod username --addattr=objectClass=ruPostMailAccount
```

Для службы каталогов Active Directory расширять схему указанным классом и дополнительными атрибутами не требуется.

11.2.2. Ключевые атрибуты

Ключевыми атрибутами учётных записей LDAP у пользователей RuPost являются:

- «mail»: в этот атрибут при заведении пользователя записывается первичный почтовый псевдоним пользователя. На этот атрибут опирается процедура аутентификации в LDAP пользователя компонентами RuPost.

Внимание!
Любое изменение и правка указанного атрибута для существующего в системе RuPost пользователя приведёт к неработоспособности связанного с ним почтового аккаунта! Этим атрибутом должна управлять только система RuPost.

- «proxyAddresses»: в этот атрибут записывается первичный почтовый адрес пользователя, а также его дополнительные псевдонимы в синтаксисе аналогичном одноимённому атрибуту в Active Directory. Запись в этот атрибут системой RuPost осуществляется при обновлении свойств почтового ящика

(нажатии на кнопку «Сохранить» в окне свойств почтового ящика - на странице «Получатели → Почтовые ящики»), а также при создании нового почтового ящика, кроме случая добавления ящика с опцией «Импорт первичных почтовых адресов из LDAP». В этом случае атрибут «proxyAddresses» записи LDAP не изменяется.

Если атрибут «proxyAddresses» в учётной записи LDAP отсутствует, или в него не записан первичный почтовый адрес, авторизация пользователя будет невозможна. Соответственно, задача атрибута «proxyAddresses» состоит не только в сообщении внешним системам всех почтовых псевдонимов пользователя, но и для подтверждения корректности выбора RuPost учетной записи LDAP для авторизации - в том случае, когда в LDAP есть несколько записей с одинаковым значением атрибута mail и только в одной из них указан первичный почтовый адрес в атрибуте «proxyAddresses».

- «objectGUID/ipaUniqueID (в зависимости от службы каталогов)»: RuPost опирается на данный атрибут для контроля актуальности ФИО, логина, подразделения и др. пользовательской информации.
- «uid/sAMAccountName (в зависимости от службы каталогов)»: логин учётной записи. Применяется для поиска учётных записей, автоматического формирования имени первичного почтового псевдонима нового ящика, в процессе миграции ящиков из системы электронной почты Microsoft Exchange в RuPost.

Указанные далее атрибуты используются для формирования глобальной адресной книги (GAL), автоматически подключаемой всем пользователям почтовой системы RuPost.

11.3. Атрибуты для глобальной адресной книги

11.3.1. FreeIPA

Имя attributeTypes	**Функциональное назначение**
cn	Полное имя
givenName	Имя
l	Город
mail	Первичный адрес электронной почты
mobile	Мобильный телефон
sn	Фамилия
st	Регион
street	Улица
title	Должность
employeeNumber	Дополнительный номер
employeeType	Тип пользователя
facsimileTelephoneNumber	Факс

ou	Департамент
pager	Пейджер
proxyAddresses	Электронная почта

11.3.2. ALD Pro

Имя attributeTypes	**Функциональное назначение**
cn	Полное имя
givenName	Имя
l	Город
mail	Первичный адрес электронной почты
mobile	Мобильный телефон
sn	Фамилия
st	Регион
street	Улица
title	Должность
c	Страна
employeeNumber	Дополнительный номер
proxyAddresses	Электронная почта
rbtadp	Департамент
rbtamiddlename	Отчество
telephoneNumber	Рабочий телефон

11.3.3. Active Directory

Имя attributeTypes	**Функциональное назначение**
cn	Полное имя
givenName	Имя
l	Город
mail	Первичный адрес электронной почты
mobile	Мобильный телефон
sn	Фамилия
st	Регион
street	Улица
title	Должность
company	Компания

department	Департамент
facsimileTelephoneNumber	Факс
homePhone	Домашний телефон
pager	Пейджер
proxyAddresses	Электронная почта

11.3.4. Samba DC

Имя attributeTypes	**Функциональное назначение**
Cn	Полное имя
givenName	Имя
l	Город
mail	Первичный адрес электронной почты
mobile	Мобильный телефон
sn	Фамилия
st	Регион
street	Улица
title	Должность
company	Компания
department	Департамент
facsimileTelephoneNumber	Факс
homePhone	Домашний телефон
pager	Пейджер
proxyAddresses	Электронная почта

11.3.5. Avanpost DS

Имя attributeTypes	**Функциональное назначение**
cn	Полное имя
givenName	Имя
l	Город
mail	Первичный адрес электронной почты
mobile	Мобильный телефон
sn	Фамилия
st	Регион

street	Улица
title	Должность
company	Компания
department	Департамент
facsimileTelephoneNumber	Факс
homePhone	Домашний телефон
pager	Пейджер
proxyAddresses	Электронная почта

12. Приложение 3. Сетевые настройки (порты)

На узле с установленным ПО Rupost должны быть открыты следующие входящие порты:

Порт	Протокол	Источник	Описание
tcp/25	SMTP (STARTTLS)	Интернет, интранет	Входящий почтовый трафик от других серверов (и/или от Relay сервера)
tcp/465	SMTPS	Интернет, интранет	SMTPS (SMTP Secure) Входящий почтовый трафик от клиентов
tcp/993	IMAPS	Интернет, интранет	IMAPS (IMAP Secure) доступ к электронной почте
tcp/995	POP3S	Интернет, интранет	POP3S (POP3 Secure) доступ к электронной почте по протоколу POP3
tcp/4190	SIEVE-TLS	Интернет, интранет	работа с пользовательскими фильтрами
tcp/80	HTTP	Интернет, интранет	301 код возврата (Moved Permanently) на 443 порт HTTPS, незащищенный Autodiscovery
tcp/443	HTTPS	Интернет, интранет	WEB клиенты, защищенный Autodiscovery (сайт и геокластер)
tcp/5000	HTTP/HTTPS	Геокластер, интранет	Панель управления RuPost (сайт и геокластер)
tcp/32000	HTTP	Интранет	дополнительные проверки Healthcheck в кластере
tcp/11211	TCP	Интранет	трафик Memcached (кэширование данных rupost-mua, состояние кэша rupost-

			dcp)
tcp/40024	HTTPS	Геокластер,, интранет	WEB клиенты, защищенный Autodiscovery
tcp/40466	SMTP	Геокластер,, интранет	Пересылка почты между сайтами (сайт и геокластер)
tcp/40993	IMAP	Геокластер,, интранет	Доступ к электронной почте (сайт и геокластер)
tcp/44190	HTTPS	Геокластер,, интранет	WEB клиенты, защищенный Autodiscovery