



## Почтовая система RuPost

---

### Руководство по установке и конфигурированию

RU.47022019.10001-01 91 01

© 2021-2023, ООО «РyПост» (RuPost, LLC.). Все права защищены.

РyПост, RuPost, WorksPad, логотип WorksPad являются торговыми марками или зарегистрированными торговыми марками РyПост (RuPost, LLC.) в США, России и других странах.

Названия прочих компаний и продуктов, упомянутые здесь, могут являться товарными знаками соответствующих компаний.

Продукты сторонних фирм упоминаются исключительно в информационных целях и конфигурирования зависимостей RuPost. Компания РyПост не несет ответственности за эксплуатационные качества и использование этих продуктов. Все договоренности, соглашения или гарантийные обязательства, при наличии таковых, заключаются непосредственно между поставщиком и потенциальными пользователями. При составлении данного руководства были предприняты все усилия для обеспечения достоверности и точности информации. Данное руководство является предметом изменений в соответствии с динамикой развития продукта и может не содержать наиболее последних версий копий экранов, имен параметров и других характеристик продукта.

Официальный веб-сайт: <http://www.rupost.ru> .

## Оглавление

1. Обзор системы .....	5
1.1. Функциональные возможности RuPost .....	5
1.2. Архитектура и компоненты сервера RuPost .....	6
1.3. Отказоустойчивый кластер RuPost .....	9
2. Подготовка к установке RuPost .....	13
2.1. Системные требования .....	13
2.2. Операционная система .....	13
2.3. Синхронизация времени .....	14
2.4. Службы каталогов LDAP .....	14
2.4.1. Служебная учётная запись .....	15
2.4.2. Интеграция с ALD Pro .....	15
2.4.3. Подготовка FreeIPA .....	16
2.4.4. Подготовка Microsoft Active Directory .....	16
2.5. Система управления базами данных .....	18
2.5.1. Настройка PostgreSQL .....	18
2.5.2. Настройка Tantor SE .....	19
2.6. Служба кэширования объектов в оперативной памяти Memcached .....	19
2.7. Подключение сетевых каталогов файловой системы NFSv4 .....	20
2.7.1. Рекомендации по настройке сетевого файлового хранилища на примере NFS сервера, входящего в состав ОС Astra Linux 1.7 .....	21
2.8. Настройки DNS .....	22
3. Установка RuPost .....	25
3.1. Установка системы с помощью мастера установки .....	25
3.2. Командный интерфейс мастера установки (CLI) .....	28
3.3. Обновление системы .....	28
3.4. Подготовка системы к реальной эксплуатации (меры информационной безопасности) .....	29
3.4.1. Генерация устойчивого уникального ключа Диффи-Хеллман .....	29
3.4.2. Использование действительных корпоративных сертификатов .....	29
3.5. Действия после установки и настройка системы .....	31
3.6. Удаление RuPost из операционной системы .....	32
3.7. Основные пути и файлы системы .....	32
4. Средства диагностики .....	34

---

4.1. Единый сводный журнал (лог) для всех почтовых компонентов – команда CLI logs.....	34
4.2. Поддержка сбора и экспорта логов – команда CLI report.....	34
4.3. Поддержка SOSReport .....	34
Приложение 1. Расчёт системных требований в зависимости от планируемой нагрузки.....	35
Общие замечания к расчёту минимальных системных требований .....	35
Оперативная память.....	35
Процессор .....	36
Дисковая память.....	36
Подключения к базе данных PostgreSQL .....	36
Пример расчёта минимальных системных требований .....	37
Приложение 2. Функциональное взаимодействие RuPost с подключенными доменами LDAP .....	38
1. Права доступа к атрибутам у служебной учётной записи RuPost .....	38
1.1 . FreeIPA .....	38
1.2 . ALD Pro .....	38
1.3 . Active Directory .....	39
2. Функциональное использование объектных классов и атрибутов LDAP .....	40
2.1 . Классы.....	40
2.2 . Ключевые атрибуты.....	40
2.3 . Атрибуты для глобальной адресной книги .....	41
2.3.1 . FreeIPA .....	41
2.3.2 . ALD Pro .....	41
2.3.3 . Active Directory .....	42
Приложение 3. Сетевые настройки (порты) .....	44

## 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
- Роли 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 ИБ 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
  - 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, P7)
- Поддержка безопасного 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 поставляется в двух редакциях:

- Стандартная – Standard
- Корпоративная – Enterprise

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

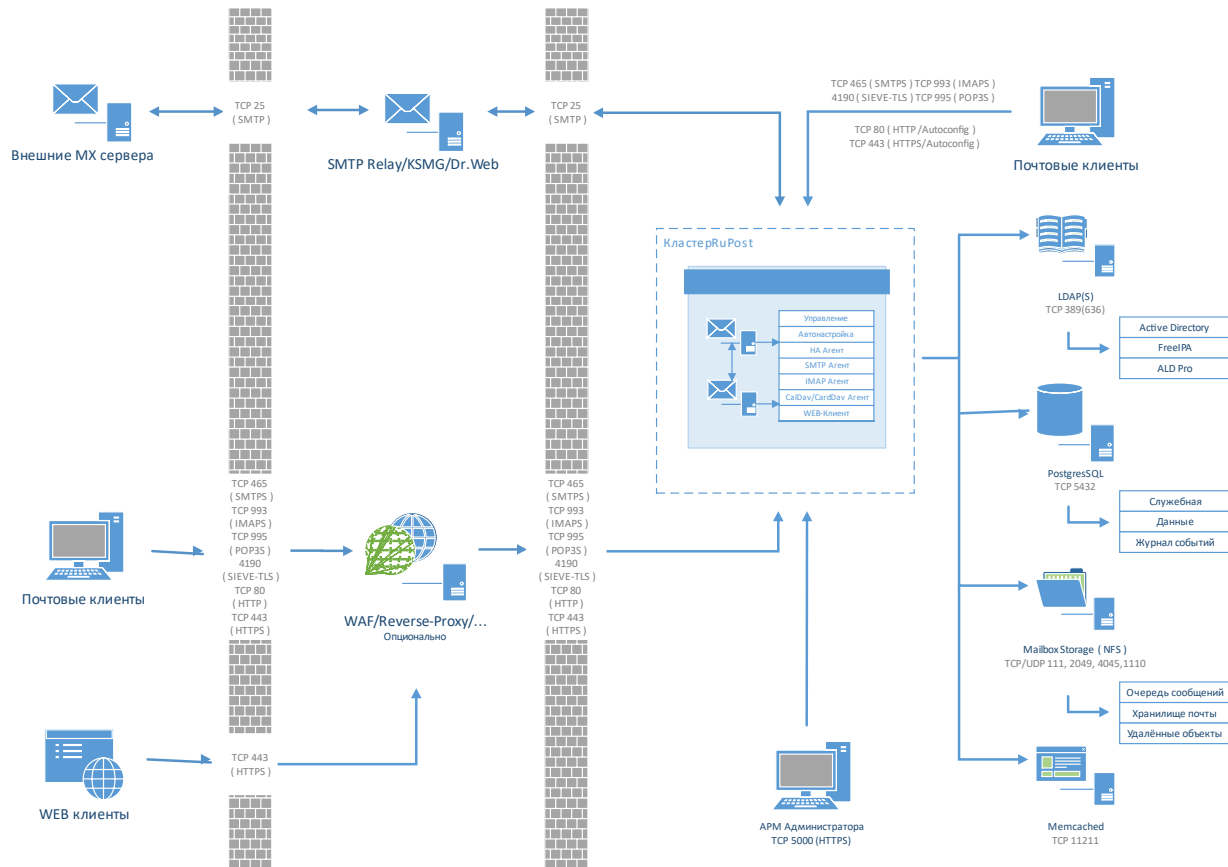
## 1.2. Архитектура и компоненты сервера RuPost

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 (APM 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) – 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;
  - безопасный доступ клиентских приложений к почтовым ящикам;
  - IMAPS, SMTPS, POP3S и безопасный доступ к web-клиенту с использованием SSL/TLS.
- **SMTP Агент – Postfix**. Компонент пересылки писем (Mail Transfer Agent, MTA), работающий по протоколам TCP, SMTP, LMTP, STARTTLS, TLS, SASL, LDAP, Milster. 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 безопасности (например, Kaspersky Security) для фильтрации входящей и исходящей почты и соединений по протоколу Milster.
- **IMAP Агент – 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;
  - выполнение функции авторизации клиентов;
  - 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.
- **CalDAV/CardDAV компонент для календарей и контактов – SOGo**. Отвечает за хранение и удалённый доступ к корпоративным и пользовательским календарям, задачам, контактам и корпоративной адресной книге. Работает по протоколам CalDAV и CardDAV.
- **Web-клиент корпоративной почты – SOGo**. Доступен во всех актуальных версиях современных браузеров.
- **Web-сервер – Nginx**. Обеспечивает работу web-клиента почты.



- **Компонент кеширования в оперативной памяти – Memcached.** Работает по протоколу TCP. Выполняет функцию кэширования и синхронизации части пользовательских данных, для увеличения быстродействия доступа к календарям, контактам и web-клиенту.

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 и ИБ 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.

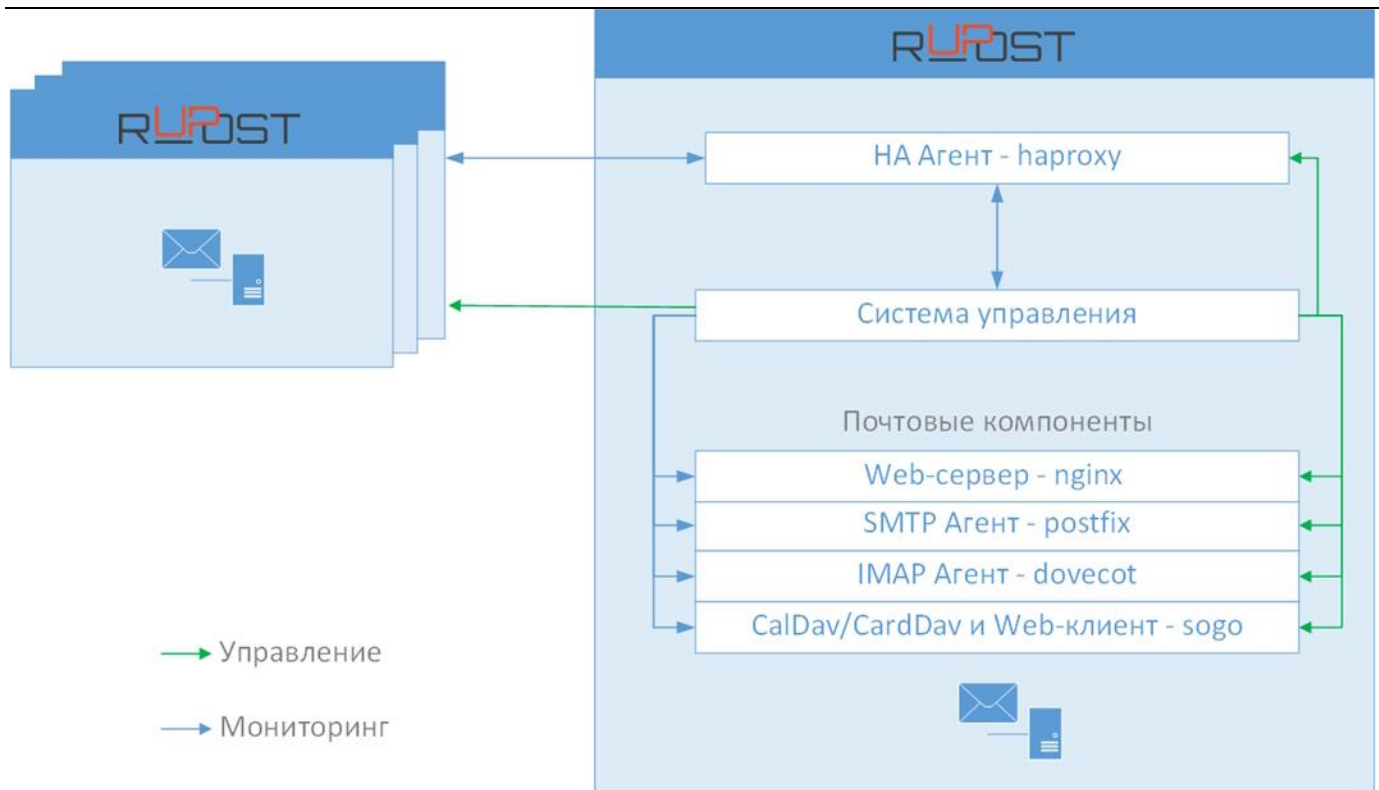
При **развертывании конфигурации** на базе выбранного шаблона система управления RuPost генерирует все необходимые конфигурационные файлы для компонентов системы.

### 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 – nginx
  - SMTP Агент – postfix
  - IMAP Агент – dovecot
  - CalDav/CardDav сервер с входящим в него почтовым Web-клиентом системы – sogo

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



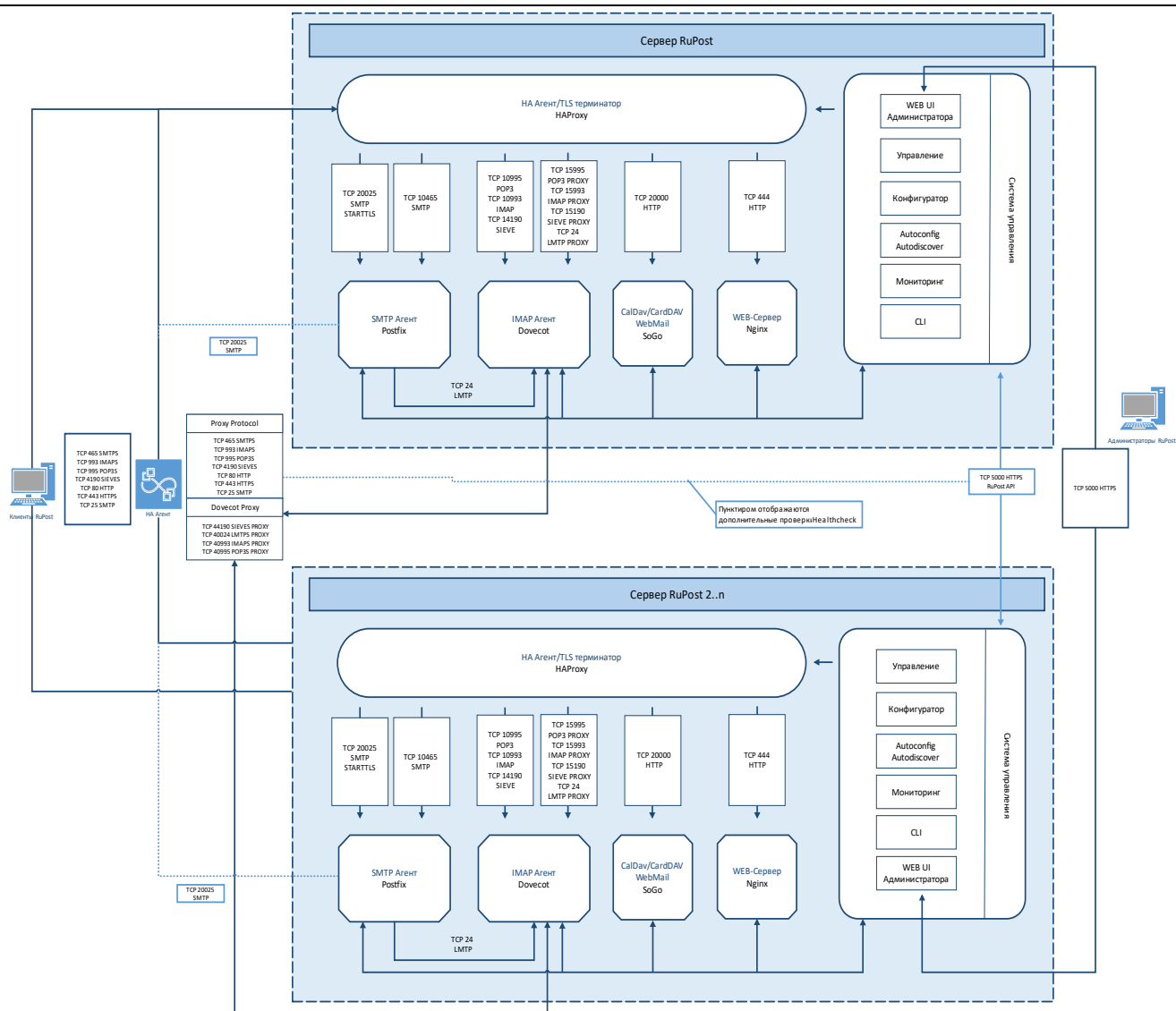
**Экземпляр системы введен в эксплуатацию**, то есть является функционирующим элементов кластера, при двух условиях:

- для всех компонентов успешно развернута активная конфигурация системы
- все компоненты запущены и функционируют 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 А 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!**

1. 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.
2. 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 производиться только при условии обеспечения одинаковой скорости доступа к LDAP, СУБД и системе хранения.

## 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 100 необходимо устанавливать PostgreSQL и Memcached только на выделенных для них отдельных узлах. При одноузловом развёртывании с менее 100 пользователей PostgreSQL и Memcached можно устанавливать локально на этот же узел, где разворачивается RuPost.

### 2.2. Операционная система

Версия RuPost 2.2.1 поддерживает ОС **Astra Linux Special Edition (ALSE) 1.7** – 1.7.1, 1.7.2, 1.7.3, 1.7.4 и их оперативные обновления.

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

Операционная система	Установочный пакет RuPost
Astra Linux Special Edition 1.7.4 и выше	rupost-2.5.0-amd64.run

При установке **Astra Linux** необходимо выбрать следующие опции:

Средства работы в Интернет

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

```
autotools-dev gobjc liblasso3 libobjc4 libxmlsec1-openssl libxmlsec1 lua-json lua-lpeg
```

#### 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.

## 2.4. Службы каталогов LDAP

Почтовая система RuPost использует домены LDAP для авторизации пользователей. Одновременно к системе RuPost может быть подключено несколько независимых доменов LDAP. Завести почтовые ящики можно только для имеющих 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

- FreeIPA
- Microsoft Active Directory

#### 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 не требует расширения схемы домена для работы с RuPost (версии ALD Pro 1.3.x и выше).

Для интеграции с RuPost в ALD Pro необходимо завести служебную учётную запись. Для этого на контроллере домена ALD Pro (права суперпользователя необязательны) получите Kerberos билет администратора домена. Это можно сделать, выполнив команду и введя пароль администратора:

```
kinit admin
```

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

```
python3 /opt/rbta/ad/mgmtportal/api/core/manage.py rupostadmin find
```

Если в выводе записи отсутствуют, или необходима новая, для служебной записи **ldapbind** с паролем **12345678** команда будет иметь следующий вид:

```
python3 /opt/rbta/ad/mgmtportal/api/core/manage.py rupostadmin add --uid ldapbind --password 12345678
```

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
```

Утилита **rupostadmin** поддерживает и другие команды управления сервисными учётными записями RuPost.

##### Общая схема работы с утилитой:

```
python3 /opt/rbta/ad/mgmtportal/api/core/manage.py rupostadmin command
```

##### 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 "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  
"-", "\_", "\$", "."

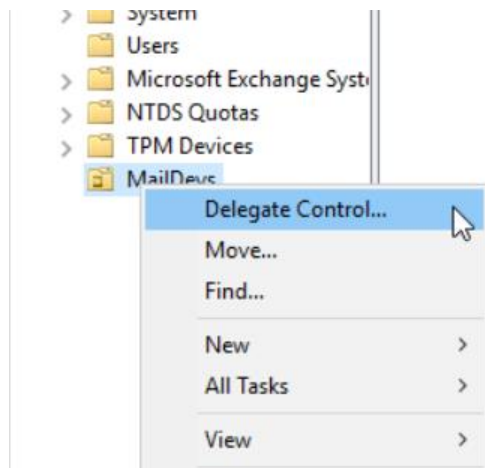
-p [PASSWORD], --password [PASSWORD] Пароль УЗ RuPost: может быть от 7 до 255 символов,  
не может начинаться или заканчиваться пробелом,  
не может содержать символы "`", "~", "{", "}", "'", '"', ""

### 2.4.3. Подготовка FreeIPA

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

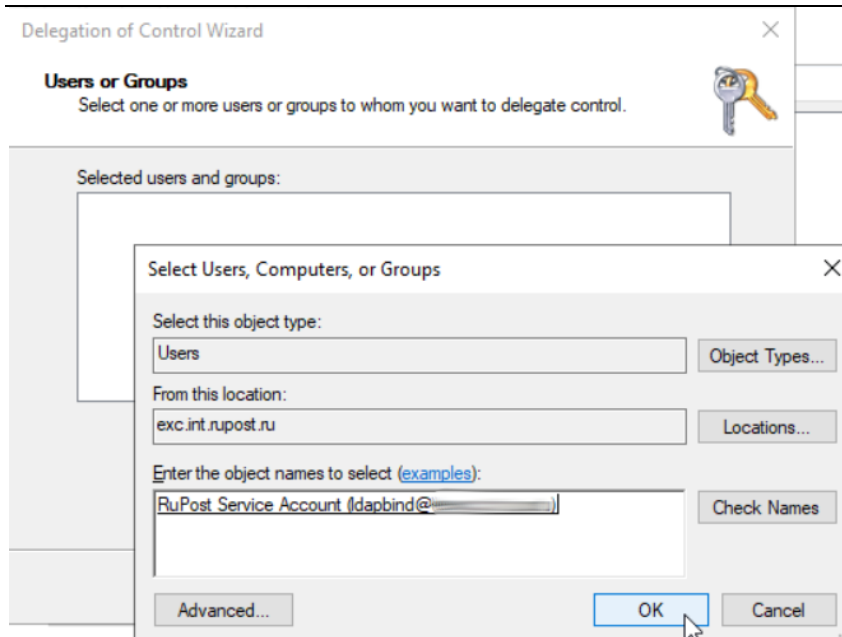
### 2.4.4. Подготовка Microsoft Active Directory

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...



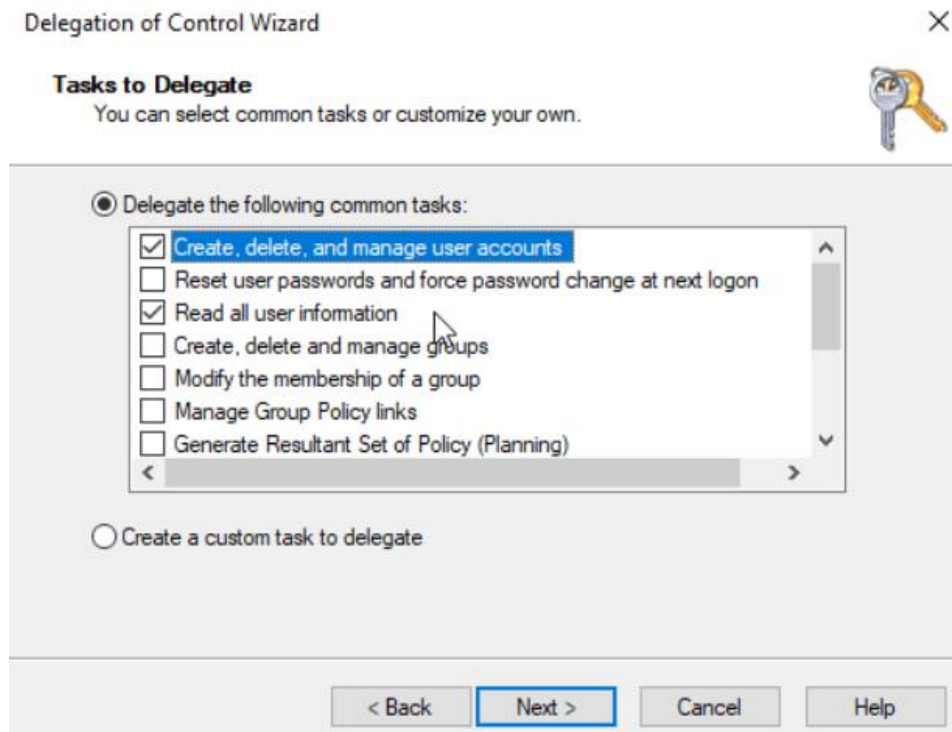
Далее необходимо добавить сервисную учётную запись RuPost для расширения прав.





После этого выберите 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.

## 2.5. Система управления базами данных

Поддерживаемые СУБД:

- PostgreSQL версии не ниже 9.6, **рекомендуется версия 13**
- Tantor SE

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 AstraLinux.

### 2.5.1. Настройка PostgreSQL

В случае, если требуется подключение к серверу баз данных, развёрнутому в инфраструктуре организации, то необходимо до начала установки RuPost создать специальную роль с параметрами NOSUPERUSER NOCREATEROLE CREATEDB LOGIN ENCRYPTED PASSWORD с помощью команды:

```
CREATE ROLE <ИМЯ_РОЛИ> WITH NOSUPERUSER CREATEDB NOCREATEROLE LOGIN ENCRYPTED  
PASSWORD '<ПАРОЛЬ>';
```

Созданная роль должна быть использована в качестве пользователя базы данных.

**Рекомендуемые параметры настройки СУБД PostgreSQL для оптимального быстродействия (файл postgresql.conf):**

```
shared_buffers = 256MB  
temp_buffers = 256MB  
work_mem = 128MB  
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
```

### 2.5.2. Настройка Tantor 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
```

Создать специальную роль с параметрами NOSUPERUSER NOCREATEROLE CREATEDB LOGIN ENCRYPTED PASSWORD с помощью команды:

```
CREATE ROLE <ИМЯ_РОЛИ> WITH NOSUPERUSER CREATEDB NOCREATEROLE LOGIN ENCRYPTED  
PASSWORD '<ПАРОЛЬ>';
```

Созданная роль должна быть использована в качестве пользователя базы данных.

### 2.6. Служба кэширования объектов в оперативной памяти Memcached

Поддерживается служба Memcached версии не ниже 1.4.33.

При кластерной конфигурации или при одноузловой схеме развёртывания системы и количестве почтовых ящиков более 100 необходимо устанавливать Memcached только на выделенный узел.

#### Внимание!

В памяти сервиса Memcached хранятся пары логин/пароль от базы данных, и хеш паролей пользователей от Idap, 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) =  $512 * (\text{количество доменов LDAP}) + (10 * \text{количество пользователей})$

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 Memcached:

```
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. Подключение сетевых каталогов файловой системы NFSv4

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

- Для всех подключаемых каталогов (почтовых очередей, хранилища почтовых ящиков, хранилища индексных файлов, 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:

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

### 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/MailQueues 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/IndexFiles 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]

**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**, очередей **MailQueues** 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.

### 2.7.1. Рекомендации по настройке сетевого файлового хранилища на примере NFS сервера, входящего в состав ОС Astra Linux 1.7

В примере рассматривается конфигурация, рассчитанная для обслуживания до 20000 почтовых ящиков, при среднестатистической ежедневной нагрузке 80% на чтение и 20% на запись. При указанной нагрузке к системным требованиям к серверу относятся:

- 4 ядра CPU;
- 4 GB оперативной памяти;
- Дисковая подсистема в 250 IOPS на 10000 активных соединений.

При этой конфигурации 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
```

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 связаны 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. 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 адреса 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:**

- **А запись** для имени хоста почтовой системы (заполняется в общих настройках под заголовком «Имя узла почтовой системы в DNS MX записи»), указывающую на IP адреса шлюзов или балансиров почтовой системы, если они имеют публичные IP адреса. 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 **А 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 **А 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 адреса 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, балансировщики 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 прокси» все IP адреса балансировщиков почтовой системы, использующих **Proxy Protocol**.



### 3. Установка RuPost

**Внимание!**

Перед началом установки RuPost 2.5.0 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.

В случае кластерного развертывания, запуск мастера установки выполняется на всех узлах системы. Все узлы, входящие в кластер, должны быть обновлены – не допускается использование разных версий системы на разных узлах кластера.

После обновления всех узлов кластера, нужно подготовить NFS для хранения индексных файлов:

1. Создать папку для размещения индексных файлов на NFS сервере;
2. В «Общих настройках» -> «Почта» для хранилища индексных файлов указать выбранную папку;
3. Повторно развернуть конфигурацию.

**Внимание!**

Начиная с версии 2.5.0 установка системы производится с помощью нового мастера установки `rupost-2.5.0-amd64.run`

**Рекомендуется**

Обновить операционную систему на узлах RuPost и NFS до версии AstraLinux 1.7.4 и более.

#### 3.1. Установка системы с помощью мастера установки

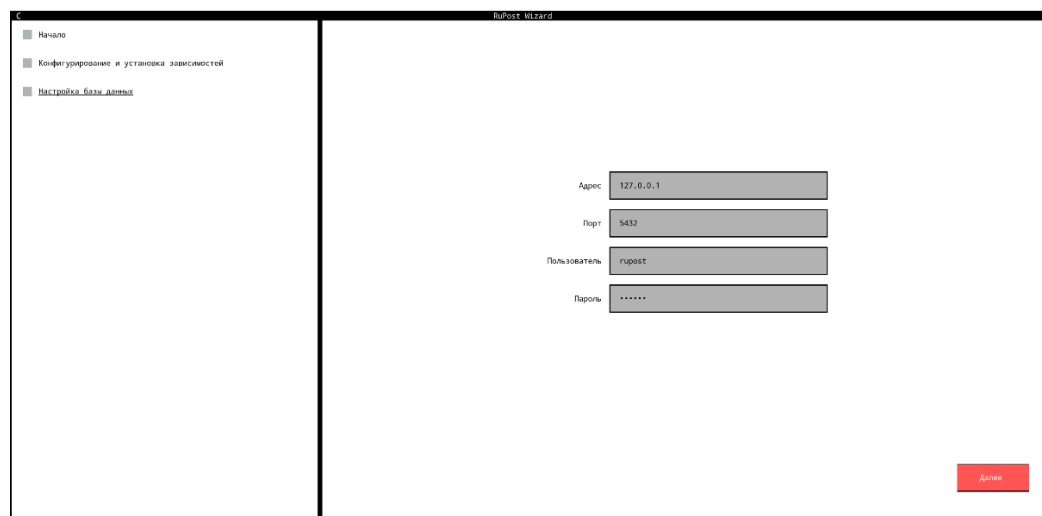
Для установки необходимо выполнить следующую команду:

```
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.

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.

### 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.

Флаги вызова:

`--help` вывести данное сообщение и выйти  
`--silent` провести автоконфигурацию в "тихом" режиме, без использования графического конфигуратора

Аргументы, используемые при флаге `--silent`:

`-h, --db-host HOST` (или `-h HOST`), где `HOST` = адрес для подключения к СУБД  
`-p, --db-port PORT` (или `-p PORT`), где `PORT` = порт для подключения к СУБД  
`-n, --db-name NAME` (или `-n NAME`), где `NAME` = имя базы данных RuPost (`rupost`)  
`-d, --data-db-name NAME` (или `-d NAME`), где `NAME` = имя базы пользовательских данных (`rupost_data`)  
`-l, --logs-db-name NAME` (или `-l NAME`), где `NAME` = имя базы данных журналов (`rupost_logs`)  
`-u, --db-user USERNAME` (или `-u USERNAME`), где `NAME` = имя пользователя для подключения к СУБД  
`--db-password PASSWORD`, где `PASSWORD` = пароль для подключения к СУБД (может быть передан и через переменную окружения `RUPOST_INSTALLER_DB_PASSWORD`). Пароль должен быть заключен в одинарные кавычки.  
`-- skip-cluster-check` – флаг пропуска проверки работающих экземпляров в кластере. По-умолчанию = `False`.

#### Внимание!

Для передачи аргументов командной строки при установке, нужно обязательно добавлять два тире `--` после `rupost-2.5.0-amd64.run` и перед любыми передаваемыми аргументами.

Пример вызова:

```
sudo sh rupost-2.5.0-amd64.run -- --silent --db-host 127.0.0.1 --db-port 5432  
--db-user rupost --db-password 'rupost' --db-name rupost --data-db-name  
rupost_data --logs-db-name rupost_logs
```

### 3.3. Обновление системы

#### Внимание!

При обновлении системы с предыдущей версии требуется:

1. перед обновлением выполнить резервное копирование узла, на котором развернут(ы) RuPost;

2. установить новую версию на всех узлы системы;
3. повторно развернуть активную конфигурацию с использованием обновленной версии необходимого шаблона конфигураций.

При обновлении системы нет необходимости останавливать процессы RuPost – они будут автоматически остановлены после обновления с сохранением всех конфигурационных параметров.

### 3.4. Подготовка системы к реальной эксплуатации (меры информационной безопасности)

#### Внимание!

Совокупность представленных ниже мер требуется для снижения рисков информационной безопасности.

#### 3.4.1. Генерация устойчивого уникального ключа Диффи-Хеллман

Данное действие необходимо для обеспечения прямой секретности SSL-соединений (Forward Secrecy). Генерируемый ключ не должен использоваться для получения каких-либо дополнительных ключей.

```
openssl dhparam -out rupost-builtin-dhparam.pem 4096
```

Добавление производится вызовом командного интерфейса RuPost CLI:

```
rupost dhparam import /путь/rupost-builtin-dhparam.pem
```

Достаточно сделать это один раз, так как в конфигурационной базе данных RuPost хранится только один экземпляр ключа, и он автоматически распространяется на все узлы системы при развёртывании конфигурации.

#### Внимание!

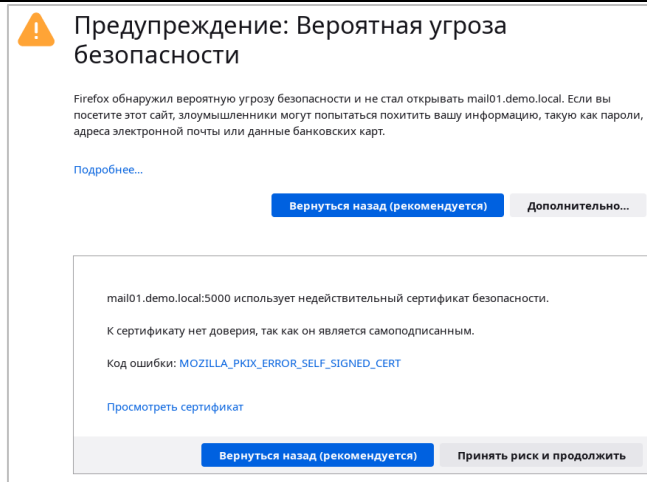
После обновления ключа необходимо повторно развернуть активную конфигурацию.

#### 3.4.2. Использование действительных корпоративных сертификатов

#### Внимание!

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

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



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" \  
  --login mailadmin \  
  --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
- mailadmin - имя администратора от имени которого происходит загрузка
- 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" \  
  --login mailadmin \  
  --cert-type control_panel \  
  --cert-path certs/rupost-control-panel.crt \  
  --key-path certs/rupost-control-panel.key
```

**Внимание!**

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.

### 3.5. Действия после установки и настройка системы

Дальнейшая настройка и управление RuPost осуществляется через командный интерфейс или в графической **Панели управления** RuPost, описанном 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
```

```
aadmin@mail01:~$ sudo rupost --help
Описание:
  RuPost CLI

Параметры:
  --help  Выводит данную подсказку

Команды:
  about          Вывести краткую сводку о приложении
  admins         Группа команд для управления администраторами
  audit          Вывести лог активности администраторов
  cert           Группа команд для управления сертификатами
  components     Группа команд для управления почтовыми компонентами
  db             Группа команд для управления базой данных
  distribution-lists  Группа команд для управления почтовыми списками рассылки
  impersonation  Группа команд для управления аккаунтом имперсонации
  instances      Группа команд для управления экземплярами системы
  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 ящиками
  mailqueues     Группа команд для управления почтовыми очередями
  make-gal       Выполнить обновление корпоративной адресной книги
  push           Группа команд для управления push уведомлениями
  report         Собрать всю информацию о работе экземпляра системы
  resources      Группа команд для управления ресурсами календаря
  restrictions   Группа команд для управления почтовыми ограничениями
  run            Запустить приложение
  sieve          Группа команд для настройки работы с глобальными Sieve скриптами
  template       Группа команд для управления шаблонами конфигураций
```

**Внимание!**

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

### 3.6. Удаление RuPost из операционной системы

Для удаления RuPost выполните команду

```
sudo apt remove 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”*).

### 3.7. Основные пути и файлы системы

Ключевые файлы и директории, 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 dovecot и 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 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 SOGo располагается по пути `/var/log/sogo/sogo.log`.
- Лог прокси-сервера 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`.
- Лог компонента HAProxy доступен в файле `/var/log/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`

## 4. Средства диагностики

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

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
```

### 4.2. 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

### 4.3. 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]
```

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

### Общие замечания к расчёту минимальных 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 минимальным системным требованиям необходимо добавить занимаемые операционной системой ресурсы, а также зарезервировать 15% 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. Кроме того, указанный далее расчёт системных требований является минимально допустимым.

По возможности, ресурсы желательно резервировать с некоторым дополнительным запасом на случай выведения из строя какого-либо узла. В подобных ситуациях обслуживание пользователей с утраченного узла автоматически продолжится активными узлами кластера RuPost, увеличив на них нагрузку. Чем больше узлов в кластере, тем меньше ресурсов необходимо резервировать на обеспечение обслуживания пользователей с выведенного из эксплуатации узла.

### Оперативная память

Вне зависимости от числа пользователей, обслуживаемых почтовой системой, на каждом узле RuPost резервируется 1 ГБ. Для каждого обслуживаемого пользователя, который использует WEB интерфейс для доступа к своим ящикам, календарям и контактам, прибавляется по 45 МБ к указанному объёму. Если пользователи не используют возможности WEB интерфейса, а получают доступ к своим ящикам, календарям и контактам через настольные клиенты, например, RuPost Desktop, или через мобильное рабочее место WorksPad, расчёт памяти на одного пользователя снижается до 35 МБ. Соответственно, при минимальном использовании WEB интерфейса каждый отдельный узел кластера RuPost сможет обслужить больше пользователей, чем в случае получения ими доступа к почте через браузер.

Так как важнейшим фактором для определения потребления оперативной памяти является учёт числа пользователей, получающих доступ к своим ящикам, календарям и контактам через WEB интерфейс, этому следует уделить особое внимание. В составе почтовой системы RuPost поставляются расширенные встроенные шаблоны, содержащие раздел «Настройка WEB клиента». В нём требуется ввести значения для двух параметров. Первый, «Число обработчиков WEB клиента», заполняется из расчёта по следующей формуле:

Число обработчиков WEB клиента = Общее число пользователей, регулярно использующих WEB интерфейс / 3 (три) / число узлов почтовой системы

Однако, у параметра «Число обработчиков WEB клиента» имеется важный нюанс. Даже если ни один из пользователей не будет получать доступ к своему ящику через WEB интерфейс, значение этого параметра не допускается менее 20.

Что касается параметра «Лимит оперативной памяти одного обработчика», то для большинства установок значение по умолчанию, равное 256, удовлетворительно для стабильной работоспособности системы. В ряде случаев, например, когда корпоративная адресная книга и её заполненность всевозможными атрибутами имеет значительный размер, а также когда система настроена на отправку писем размером более 100 МБ, возможно, потребуется увеличение этого параметра до значений 384 или 512. Это значение определяется эмпирически, исходя из потребностей конкретной установки почтовой системы, но не должно быть менее 256.

## Процессор

Минимальные системные требования по процессору вычисляются по формуле:

На каждые 100 пользователей 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:  $750 \text{ (общее число пользователей)} / 100 \text{ (сто)} / 2 \text{ (узла в кластере)} = 4$  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.

## 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.

## Подключения к базе данных PostgreSQL

Каждый узел кластера RuPost в среднем использует фиксированное число подключений, равное 90. Сверх этого каждый узел использует для каждого обработчика WEB клиента по два соединения. Общее число соединений можно получить по формуле:

Общее число соединений =  $(90 + \text{число обработчиков WEB клиента на одном узле} * 2) * \text{число узлов в кластере RuPost} + 15\%$  (обязательный резерв)

Для некоторых операций число соединений временно может превышать расчётные, поэтому резерв в 15% в этом случае обязателен.

### Пример расчёта минимальных системных требований

Предположим, что в организации необходимо обеспечить электронной почтой, календарями и корпоративными контактами 1400 пользователей. Для отказоустойчивости, система будет установлена на трёх узлах. В числе требований — ожидается регулярное единовременное обращение всех пользователей организации к почтовому WEB интерфейсу. Соответственно, минимальная свободная оперативная память на каждом узле, без учёта потребления ресурсов операционной системой, должна составлять:

$$1400 \text{ (пользователи)} / 3 \text{ (узлы)} * 45 \text{ МБ (при использовании WEB интерфейса)} + 1 \text{ ГБ (константа)} + 15\% \text{ (обязательный резерв)} + 20\% \text{ (на случай отказа одного из узлов)} = 29720 \text{ МБ}$$

При разворачивании конфигурации в разделе «Настройка WEB клиента» параметру «Число обработчиков WEB клиента» следует присвоить значение:

$$1400 \text{ (пользователи)} / 3 \text{ (три)} / 3 \text{ (узлы)} = 156 \text{ обработчиков (округление в большую сторону)}$$

В том же разделе параметру «Лимит оперативной памяти одного обработчика» следует указать значение по умолчанию 256. Возможно, по опыту эксплуатации почтовой системы это значение придётся увеличить.

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

$$1400 \text{ (пользователи)} / 100 \text{ (сто)} / 3 \text{ (узлы)} + 15\% \text{ (обязательный резерв)} + 20\% \text{ (на случай отказа одного из узлов)} = 7 \text{ ядер (округление в большую сторону)}$$

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

$$(90 \text{ (фиксированное число)} + 156 \text{ (число обработчиков WEB клиента на одном узле)} * 2) * 3 \text{ (узлы)} + 15\% \text{ (обязательный резерв)} = 1387 \text{ (общее число подключений от кластера к базе данных)}$$

На дисковом пространстве обеспечены свободные 30 ГБ на каждом узле. На сетевой файловой системе ожидается суммарно 5 ТБ почтовых данных.

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

### 1. Права доступа к атрибутам у служебной учётной записи RuPost

#### 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

#### 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

### 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

## 2. Функциональное использование объектных классов и атрибутов LDAP

### 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 роли:

- 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.

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

### 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 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 псевдоним пользователя, а также его дополнительные псевдонимы в синтаксисе аналогичном для одноимённого атрибута в Active Directory. Запись в этот атрибут системой RuPost осуществляется только при заведении для



учётной записи LDAP почтового ящика. В дальнейшем целостность этого атрибута не проверяется, а также он не участвует в mail flow и других критично важных функциях почтовой системы. Основная задача атрибута состоит в сообщении внешним системам всех почтовых псевдонимов пользователя.

- «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.

## 2.3 . Атрибуты для глобальной адресной книги

### 2.3.1 . 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	Роль
facsimileTelephoneNumber	Факс
ou	Департамент
pager	Пейджер
proxyAddresses	Электронная почта

### 2.3.2 . 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

### 2.3.3 . 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
----------------	-------------------

## 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 Autodiscover
tcp/5000	HTTP	Интранет	Панель управления RuPost
tcp/10000	TCP	Интранет	Связь между узлами кластера
tcp/20025	SMTP	Интранет	дополнительные проверки Healthcheck в кластере