



МойОфис Частное Облако 2

В ВАРИАНТЕ ИСПОЛНЕНИЯ ГОСТ

Архитектура

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
«МОЙОФИС ЧАСТНОЕ ОБЛАКО 2»
В ВАРИАНТЕ ИСПОЛНЕНИЯ ГОСТ**

АРХИТЕКТУРА

2.8G

На 25 листах

**Москва
2024**

МойОфис

Все упомянутые в этом документе названия продуктов, логотипы, торговые марки и товарные знаки принадлежат их владельцам.

Товарные знаки «МойОфис» и «MyOffice» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

Ни при каких обстоятельствах нельзя истолковывать любое содержимое настоящего документа как прямое или косвенное предоставление лицензии или права на использование товарных знаков, логотипов или знаков обслуживания, приведенных в нем. Любое несанкционированное использование этих товарных знаков, логотипов или знаков обслуживания без письменного разрешения их правообладателя строго запрещено.

СОДЕРЖАНИЕ

1 Назначение	6
2 Общая компонентная схема	7
2.1 Состав компонентов СО	8
2.2 Состав компонентов PGS	9
3 Описание архитектуры	10
3.1 Архитектурная схема СО	10
3.1.1 Описание архитектуры СО	11
3.2 Архитектурная схема PGS	12
3.2.1 Описание архитектуры PGS	13
4 Типовые схемы установки	14
4.1 Конфигурация без отказоустойчивости	14
4.2 Кластерная отказоустойчивая конфигурация	15
4.3 Требования для кластера с профилем более 2000 пользователей	16
4.4 Расчет требований IOPS	16
4.5 Типовая схема масштабирования	17
5 Техническая поддержка	18
Приложение А - Пример файла inventory СО (установка standalone)	19
Приложение Б - Пример файла inventory СО (кластерная установка)	21
Приложение В - Пример файла inventory PGS (установка standalone)	23
Приложение Г - Пример файла inventory PGS (кластерная установка)	24

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

В настоящем документе используются следующие сокращения с соответствующими расшифровками (см. Таблицу 1).

Таблица 1 — Сокращения и расшифровки

Сокращение, термин	Расшифровка и определение
AD	Microsoft Active Directory
Ansible	Система управления конфигурациями, используемая для автоматизации настройки и развертывания программного обеспечения
API	Application Programming Interface, интерфейс программирования приложений
CO	CloudOffice, Облачный Офис, общее название продукта (группы редакторов)
Docker Registry	Масштабируемое серверное приложение для хранения и распространения контейнеров Docker
Inventory	Файл ПО Ansible с перечислением ролей и их IP-адресов
PGS	Pythagoras, Система хранения данных в составе «МойОфис Частное Облако 2»
PSN	Poseidon, приложение почты, календаря и контактов («МойОфис Почта»)
REST API	Архитектурный стиль взаимодействия компонентов распределенного приложения в сети
S3 хранилище	Сервис хранения объектов, предлагаемый поставщиками облачных услуг
SPA	Одностраничное приложение (англ. single page application, SPA) — это веб-приложение или веб-сайт, использующий единственный HTML-документ как оболочку для всех веб-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript, обычно посредством AJAX.
ОС	Операционная система
Тенант	Логический объект, включающий в себя совокупность вычислительных ресурсов, репозиторий и пользователей

1 НАЗНАЧЕНИЕ

«МойОфис Частное Облако 2» в варианте исполнения ГОСТ — комплекс безопасных веб-сервисов и приложений для организации хранения, доступа и совместной работы с файлами и документами внутри компании, использующих отечественные средства криптографической защиты информации. Взаимодействие всех клиентских приложений с серверными системами осуществляется по сетевым каналам, защищенным с помощью протокола TLS с использованием отечественной криптографии.

В состав продукта входят:

- Система хранения данных для безопасного хранения корпоративных файлов и обеспечения возможностей авторизации, аутентификации и разграничения прав доступа пользователей;
- Система редактирования и совместной работы для индивидуального и совместного редактирования презентаций, текстовых и табличных документов;
- Административная панель системы хранения для управления пользователями, группами, общими папками, доменами и тенантами.

В состав продукта входят следующие приложения для работы в веб-браузерах и мобильных устройствах:

- «МойОфис Документы» — веб-приложение для организации структурированного хранения файлов, выполнения операций с файлами и папками, настройки совместного доступа;
- «МойОфис Текст» — веб-редактор для быстрого и удобного создания и форматирования текстовых документов любой сложности;
- «МойОфис Таблица» — веб-редактор для создания электронных таблиц, ведения расчетов, анализа данных и просмотра сводных отчетов;
- «МойОфис Презентация (Beta)» — веб-редактор для создания, оформления и демонстрации презентаций;
- «МойОфис Документы» для мобильных платформ — приложение для просмотра и редактирования текстовых документов, электронных таблиц и презентаций, просмотра PDF-файлов, а также доступа к облачным хранилищам на смартфонах и планшетах с ОС Android, iOS и iPadOS.

Подробное описание возможностей продукта приведено в документе «"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Функциональные возможности».

2 ОБЩАЯ КОМПОНЕНТНАЯ СХЕМА

На рисунке 1 представлена общая схема взаимодействия компонентов «МойОфис Частное Облако 2» в варианте исполнения ГОСТ.

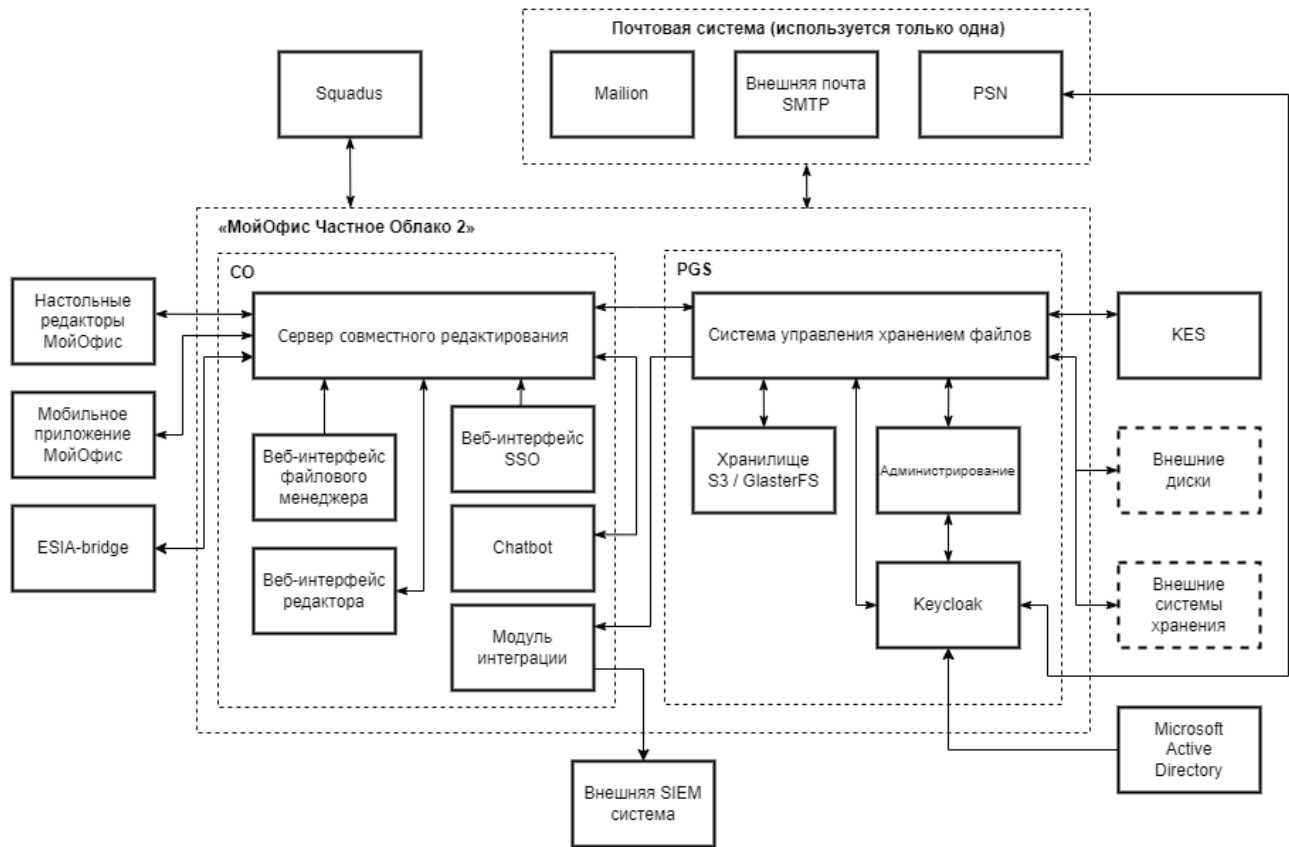


Рисунок 1 — Общая компонентная схема «МойОфис Частное Облако 2»

2.1 Состав компонентов СО

Состав и назначение компонентов Сервера совместного редактирования (СО) приведены в таблице 2.

Таблица 2 — Состав и назначение компонентов СО

Наименование компонента	Описание
Сервер совместного редактирования	Серверный компонент, обеспечивающий: <ul style="list-style-type: none">– совместное редактирование документов в Облаке с использованием любой платформы (веб, мобильные или настольные приложения);– взаимодействие клиентских приложений с Системой хранения данных (PGS);– интеграцию с почтовыми системами
Веб-интерфейс файлового менеджера	Веб-приложение «МойОфис Документы» предназначено для организации структурированного хранения файлов, выполнения операций с файлами и папками, настройки совместного доступа
Веб-интерфейс SSO	Веб-интерфейс предназначен для авторизации, аутентификации в системе и управления профилем пользователя. Главная страница доступа к приложениям (лендинг)
Веб-интерфейс редакторов	Состоит из редакторов с различным функционалом: <ul style="list-style-type: none">– веб-приложение «МойОфис Текст» — для создания и форматирования текстовых документов;– веб-приложение «МойОфис Таблица» — для создания электронных таблиц, ведения расчетов, анализа данных и просмотра сводных отчетов;– веб-приложение «МойОфис Презентация (Beta)» — для создания, оформления и демонстрации презентаций
Мобильные приложения МойОфис	Мобильное приложение «МойОфис Документы» предназначено для просмотра и редактирования текстовых документов, электронных таблиц и презентаций, просмотра PDF файлов, а также доступа к облачным хранилищам на смартфонах и планшетах с ОС Android, iOS и iPadOS. На устройствах с ОС Аврора доступен просмотр объектов хранилища
chatBot	Сервис, обеспечивающий работу виджета Squadus в окне веб-редактора
Модуль интеграции	Сервис, обеспечивающий отправку событий безопасности во внешнюю SIEM-систему

Сервер совместного редактирования (СО) предусматривает подключение пользователей настольных редакторов из комплекта приложений «МойОфис Стандартный» (подробнее см. в документе «МойОфис Частное Облако 2» в варианте исполнения ГОСТ. Руководство по настройке»).

2.2 Состав компонентов PGS

Состав и назначение компонентов Системы хранения данных (PGS) приведены в таблице 3.

Таблица 3 — Состав и назначение компонентов PGS

Наименование компонента	Описание
Система управления хранением файлов	Серверный компонент «МойОфис Частное Облако 2», обеспечивающий: <ul style="list-style-type: none"> – хранение объектов пользователя; – хранение общих корпоративных объектов; – поиск по имени и содержимому; – выполнение файловых операций для пользователя; – разграничение прав доступа; – аутентификацию и валидацию статусов учетных записей пользователей и прав доступа к системе. В качестве веб-интерфейса используется файловый менеджер
Keycloak	Сервис содержит список пользователей системы и их атрибуты. Предусмотрена возможность синхронизации пользователей из внешних каталогов (Active Directory). Keycloak обеспечивает хранение списка тенантов и их атрибутов для «МойОфис Частное Облако 2». Для управления параметрами установки от имени суперадминистратора используется графический интерфейс
Хранилище	Тип файлового хранилища определяется на этапе первичной установки компонента. Используется два типа хранилища: <ul style="list-style-type: none"> – файловая система (GlusterFS); – объектное хранилище S3 (minIO)*
Администрирование	Серверный компонент, обеспечивающий: <ul style="list-style-type: none"> – управление политиками безопасности и другими настройками тенанта (организации); – ролевую модель доступа к системе; – управление доменами, пользователями, группами пользователей, общими папками, публичными ссылками от лица администратора системы; – настройку доступности функциональных блоков «МойОфис Частное Облако 2» для пользователей. Для управления компонентом используется веб-интерфейс

* — Для хранилища типа S3 предусмотрена интеграция с другим S3 хранилищем в уже существующей инфраструктуре.

Для обеспечения уровня безопасности в PGS предусмотрена интеграция с Kaspersky Endpoint Security (KES) (подробнее см. в документе «МойОфис Частное Облако 2» в варианте исполнения ГОСТ. Руководство по настройке).

3 ОПИСАНИЕ АРХИТЕКТУРЫ

3.1 Архитектурная схема СО

На рисунке 2 представлена схема взаимодействия подсистем Системы редактирования и совместной работы (СО).

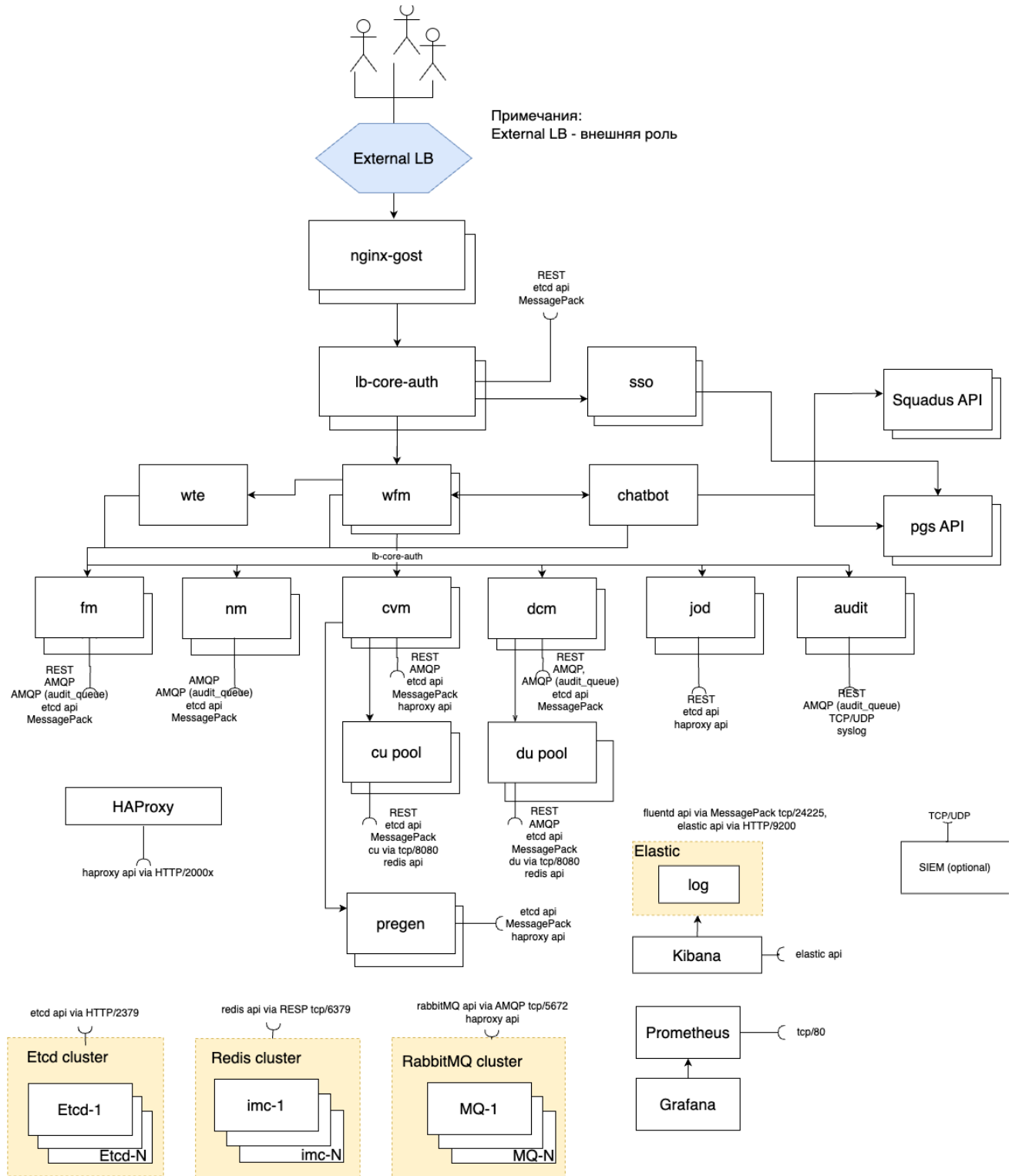


Рисунок 2 — Архитектурная схема Сервера совместного редактирования (СО)

3.1.1 Описание архитектуры СО

Описание подсистем и сервисов СО приведено в таблице 4.

Таблица 4 — Описание подсистем и сервисов СО

Наименование подсистемы или сервиса	Описание
Lb-core-auth (nginx-wfe)	Расширенный веб-сервер NGINX с поддержкой Lua. Отвечает за авторизацию, доступность API, балансировку
HAProxy	Балансировщик нагрузки между внутренними сервисами
FM	File Manager, сервис файлового менеджера. Отвечает за создание, удаление, добавление файлов, предоставление доступа для других пользователей
NM	Notification Manager, сервис управления уведомлениями.
CVM	Conversion Manager, сервис управления конвертированием файлов. При конвертации документов использует пул CU юнитов, Pregen или Jod
CU	Conversion Unit, экземпляр процесса конвертора различных форматов файлов (офисных или изображений)
Pregen	Сервис конвертации документов в pdf и json форматы
DCM	Document Collaboration Manager, сервис управления редактированием документов. При редактировании документов использует пул DU юнитов
DU	Document Unit, экземпляр процесса редактирования и коллаборации документов
JOD	Java OpenDocument Converter, сервис конвертации офисных файлов устаревших форматов (например MS Office 1997) в современные форматы или pdf
Audit	Сервис аудита, получает события аудита из сервисов и PGS через RabbitMQ. Выполняет отправку событий аудита (TCP/UDP) в syslog или SIEM (не входит в стандартную поставку)
Elastic	Elastic (Opendistro Elastic Search), поисковый и аналитический сервис. Отвечает за сбор, обработку и хранение логов всех сервисов
Fluentd	Сервис, который собирает логи от всех сервисов и передает их в Elastic Search
Kibana	Kibana, инструмент визуализации и изучения данных, анализ логов приложения, использует данные из Elastic
EtcD	Хранилище типа «ключ-значение». Отвечает за хранение свойств (настроек) всех сервисов
Redis	noSQL in-меморию хранилище, типа «ключ-значение». Отвечает за хранение кешированной информации при работе с облаком (файлами, профили пользователей, токены)
RabbitMQ	Брокер сообщений на основе протокола AMQP. Отвечает за отправку сообщений между сервисами, например за отправку уведомлений, событий аудита
LCS	License Service, сервис управления лицензиями
NPS	Native Process Service, сервис управления cu, du
Nginx-gost	Прокси-сервис, обеспечивающий доступ
SDD	Service Detector Docker, обнаружение сервисов. Опрашивает сервисы NPS через docker api и сокет
Prometheus	Инструмент сбора метрик со всех сервисов

Наименование подсистемы или сервиса	Описание
Grafana	Инструмент визуализации метрик приложения. Получает метрики из Prometheus
SSO	Single Sign-On (SPA веб-приложение). Отвечает за авторизацию пользователя, регистрацию пользователя по промокоду, страницу профиля, лендинг (список приложений в системе)
WFM	Web file manager (SPA веб-приложение). Веб-клиент файлового менеджера (Виджет вложений). Отвечает за операции с файлами (кроме редактирования) и просмотр файлов по публичной ссылке
Chatbot	NodeJs сервис. Отвечает за создание чата по документу, добавления туда пользователей, обновления списка пользователей в чате
WTE	Web text editors (Веб-клиент офисного приложения). Отвечает за редактирование и чтение файлов

3.2 Архитектурная схема PGS

На рисунке 3 представлена схема взаимодействия компонентов Системы хранения данных (PGS).

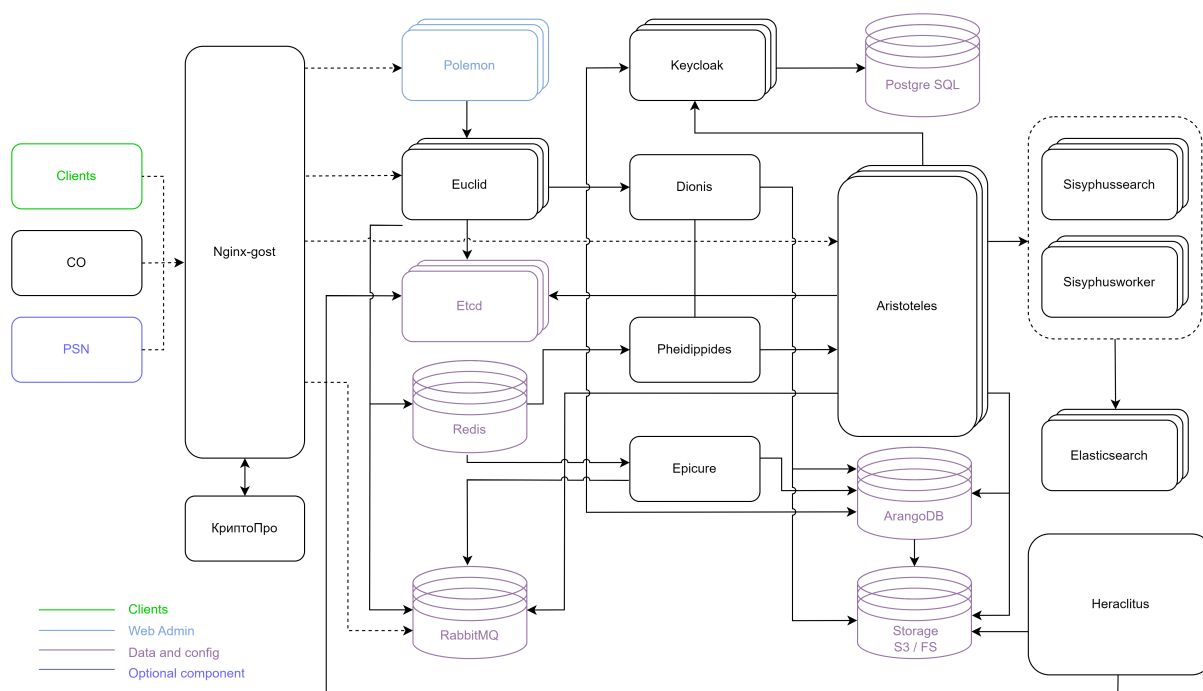


Рисунок 3 — Архитектурная схема Системы хранения данных (PGS)

3.2.1 Описание архитектуры PGS

Описание подсистем и сервисов PGS приведено в таблице 5.

Таблица 5 — Описание подсистем и сервисов PGS

Наименование сервиса	Описание
КриптоПро	Сервис, обеспечивающий шифрование при работе с документами
Arangodb	База данных, содержащая метаданные файлов (в т.ч. информацию о владельце документа, правах доступа и пр.)
Aristoteles	Сервер приложений, выступающий backend-частью для компонентов СО в части выполнения файловых операций, разграничения прав доступа, версионирования, фиксации истории событий по объектам
Dionis	Сервис, отвечающий за удаление и переназначение прав доступа для объектов пользователей
Elasticsearch	Сервис, отвечающий за поиск по содержимому по хранящимся файлам
Epicure	Сервис формирования и отправки сообщений безопасности, с последующей отправкой в аудит системы (SIEM)
Etcd	Сервис, содержащий конфигурацию приложений, при кластерном развертывании также используется сервисом Postgres для создания кластера
Euclid	Rest API сервис, отвечающий за администрирование пользователей в системе, выступающий backend-частью для компонента polemon (веб-администрирование)
Heraclitus	Сервис очистки архивных данных, удаленных пользователями из корзины. Может настраивать сроки хранения архивных данных и автоудаления их с диска по заданному расписанию
Keycloak	Сервис SSO, хранящий в себе настройки инсталляции, данные по тенантам и пользователям
Nginx-gost	Прокси-сервис, обеспечивающий доступ до: rabbitmq, aristotels, euclid
Pheidippides	Сервис, осуществляющий обработку событий в Redis каналах (автоматическая блокировка IP-адресов/публичных ссылок)
Polemon	Сервис веб-администрирования Euclid (веб-интерфейс административной панели)
Postgres	PostgreSQL, база данных для сервиса авторизации Keycloak
RabbitMQ	Очередь сообщений. Используется для передачи документов в elasticsearch для поиска по содержимому документов и для передачи межкомпонентных уведомлений PGS>СО об изменении настроек хранилища
Redis	База данных «ключ-значение» для неперсистентных данных (в основном используется для хранения токенов и других авторизационных данных)
Sisyphus_sisyphussearch	Сервис, осуществляющий поиск по содержимому документов в elasticsearch
Sisyphus_sisyphusworker	Сервис, осуществляющий передачу файлов из rabbitMQ в elasticsearch
Storage S3/FS	Блок storage — осуществляет хранение файлов системы. В качестве хранилища FS в проекте используется GlusterFS. В качестве хранилища S3 в проекте используется minIO Для хранения данных в объеме свыше 100 ТБайт рекомендуется использовать S3

4 ТИПОВЫЕ СХЕМЫ УСТАНОВКИ

4.1 Конфигурация без отказоустойчивости

Конфигурация без отказоустойчивости может использоваться при условии предоставления отказоустойчивости на уровне виртуализации.

Данная конфигурация характеризуется тем, что все серверные роли развертываются в единственном сервере. В такой конфигурации роли устанавливаются:

- на несколько виртуальных серверов с объединением ролей;
- на несколько виртуальных серверов, где одному серверу соответствует одна роль.

Установка такого типа не требует использования подсистемы балансировки.

Аппаратные требования для установки на двух виртуальных серверах приведены в таблице 6.

В данном разделе приведены требования для развертывания системы без отказоустойчивости со следующим максимальным профилем эксплуатации:

- всего пользователей - 999;
- количество одновременно активных пользователей - 400;
- количество документов, редактируемых одновременно - 200.



Внимание:

- из-за невозможности масштабирования конфигурации без отказоустойчивости (standalone) в текущем режиме максимальный допустимый лимит пользователей ≤ 1000 ;
- требования идентичны для систем с общим количеством пользователей от 1 до 999.

Таблица 6 — Аппаратные требования для конфигурации без отказоустойчивости

Роли серверов	CPU, vCPU	RAM, Гбайт	SSD, Гбайт
operator*	1	4	50
все PGS	8	20	100
все CO	8	20	100

* — сервер с ролью `operator` рекомендуется размещать на отдельном виртуальном сервере. После установки сервер с ролью `operator` можно выключить.

4.2 Кластерная отказоустойчивая конфигурация

В кластерной отказоустойчивой конфигурации каждая критическая роль реплицируется на разных виртуальных серверах. Разные роли могут быть объединены на одном виртуальном сервере. Архитектурных ограничений по объединению ролей нет. Виртуальные серверы разносятся по разным физическим серверам или гипервизорам.

Аппаратные требования для установки приведены в таблице 7. В данном разделе приведены минимальные требования для развертывания системы в режиме кластера со следующим профилем эксплуатации:

- всего пользователей - 2000;
- количество одновременно активных пользователей - 1000;
- количество документов, редактируемых одновременно - 90.

Таблица 7 — Аппаратные требования для кластерной отказоустойчивой конфигурации

Наименование роли	Количество серверов	CPU, vCPU	RAM, Гбайт	SSD, Гбайт	HDD, Гбайт
operator	1	1	4	0	50
LB	2	2	4	50	0
core	2	12	24	100	0
infra	1	2	12	100	0
mq+imc+etcd	3	6	16	50	0
PGS-APP	2	8	8	0	70
STORAGE	2	4	12	12 055	0
STORAGE-A	1	4	12	1 550	0
PGS-BE	3	4	12	0	3 350
PGS-DB	2	16	10	1 300	0
PGS-LOG	1	4	8	0	100
PFE	2	8	16	0	200
PBE	2	8	20	0	5 250
PDB	2	8	16	100	0
PLOG	1	4	8	0	100

Рекомендации по кластерной отказоустойчивой конфигурации:

- сервер `operator` не рекомендуется устанавливать на одном виртуальном сервере с приложением, но он может быть выключен после установки;
- в этой конфигурации объединены роли `cvm/cu-pool` и `dcm/du-pool`;
- роли `log` и `PGS-LOG` не являются критическими;
- роль `LB` — внешний балансировщик;
- для кластерных ролей `etcd / mq / imc` необходимо использовать минимум 3 узла, для `etcd` рекомендуется использовать 5 узлов.

4.3 Требования для кластера с профилем более 2000 пользователей

Для кластерной установки, на которой планируется работа более 2000 пользователей, необходимо обратиться к вендору для расчета размеров серверных ресурсов и получения рекомендаций по объединению ролей.

4.4 Расчет требований IOPS

Требования IOPS к различным ролям представлены в таблице 8.

Таблица 8 — Требованиях IOPS для ролей

Наименование роли	Средние значение IOPS	Максимальное значение IOPS
APP	70	100
BE*	30	100
STORAGE	N***	N***
DB**	N***	N***
INFRASTRUCTURE	20	50

* — на данном сервере размещены следующие роли: `keycloak`, `arangodb_agent`, `redis`, `rabbitmq`, `search`, `etcd`.

** — на данном сервере размещены следующие роли: `postgres`, `arangodb`.

N*** — зависит от количества одновременно редактируемых документов.

Примерный расчет требований к дисковой подсистеме для ролей ST и DB возможен по следующей формуле:

$$\frac{\text{количество одновременно редактируемых документов}}{\text{коэффициент}} = \text{требования к IOPS}$$

Коэффициент для расчета требований представлен в таблице 9.

Таблица 9 — Коэффициент IOPS для ролей

Наименование роли	Средние значение	Максимальное значение	Примечание
STORAGE	52	9	Формула работает от 1000 одновременно редактируемых документов
DB	90	55	Формула работает от 5000 одновременно редактируемых документов

При расчете следует принять во внимание, что формула носит оценочный характер. Для возможности работы сервиса пропускная способность дисковой подсистемы должна быть не ниже среднего значения IOPS, для комфортной работы пропускная способность должна быть выше максимального требования IOPS.

4.5 Типовая схема масштабирования

Полноценное масштабирование возможно использовать только для кластерной отказоустойчивой конфигурации.

Для конфигурации (standalone) возможно использование вертикального масштабирования с учетом ограничения Docker и других системных сервисов.

Переход от конфигурации (standalone) к кластерной выполняется при полной переустановке программного продукта. В PGS переход обеспечивается с помощью предварительного резервирования баз данных.

В первую очередь для компонента СО следует масштабировать узлы кластера с ролями:

- `dcm` (влияет на количество одновременно открытых документов);
- `cvm` и `pregen` (влияет на количество конвертаций, скорость загрузки, скачивания, печати документов).

5 ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

Контактная информация службы технической поддержки ООО «Новые облачные технологии» в случае возникновения вопросов, не описанных в данном руководстве:

Адрес электронной почты: support@service.myoffice.ru

Телефон: 8-800-222-1-888

ПРИЛОЖЕНИЕ А

Пример файла inventory CO (установка Standalone)

```
all:
  children:

    co:
      children:
        co_chatbot:
          hosts:
            co-infra-1.installation.example.net:

        co_etcd:
          hosts:
            co-infra-1.installation.example.net:

        co_mq:
          hosts:
            co-infra-1.installation.example.net:

        co_cvm:
          hosts:
            co-infra-1.installation.example.net:

        co_cu:
          hosts:
            co-infra-1.installation.example.net:

        co_dcm:
          hosts:
            co-infra-1.installation.example.net:

        co_du:
          hosts:
            co-infra-1.installation.example.net:

        co_fm:
          hosts:
            co-infra-1.installation.example.net:

        co_jod:
          hosts:
            co-infra-1.installation.example.net:

        co_nm:
          hosts:
            co-infra-1.installation.example.net:

        co_pregen:
          hosts:
            co-infra-1.installation.example.net:

        co_imc:
          hosts:
            co-infra-1.installation.example.net:

        co_lb_core_auth:
          hosts:
            co-infra-1.installation.example.net:
```

```
co_infra:
  hosts:
    co-infra-1.installation.example.net:
co_setup:
  hosts:
    co-infra-1.installation.example.net:
```

ПРИЛОЖЕНИЕ Б

Пример файла inventory CO (кластерная установка)

```
all:
  children:
    co:
      children:
        co_etcd:
          hosts:
            etcd-1.installation.example.net:
            etcd-2.installation.example.net:
            etcd-3.installation.example.net:
            etcd-4.installation.example.net:
            etcd-5.installation.example.net:
        co_mq:
          hosts:
            mq-1.installation.example.net:
            mq-2.installation.example.net:
            mq-3.installation.example.net:
        co_cvm:
          hosts:
            cvm-1.installation.example.net:
            cvm-2.installation.example.net:
        co_cu:
          hosts:
            cvm-1.mcs.installation.example.net:
            cvm-2.mcs.installation.example.net:
        co_dcm:
          hosts:
            dcm-1.installation.example.net:
            dcm-2.installation.example.net:
            dcm-3.installation.example.net:
        co_du:
          hosts:
            dcm-1.installation.example.net:
            dcm-2.installation.example.net:
            dcm-3.installation.example.net:
        co_fm:
          hosts:
            fm-1.installation.example.net:
            fm-2.installation.example.net:
        co_jod:
          hosts:
            cvm-1.installation.example.net:
            cvm-2.installation.example.net:
        co_nm:
          hosts:
            nm-1.installation.example.net:
            nm-2.installation.example.net:
```

```
co_pregen:
  hosts:
    pregen-1.installation.example.net:
    pregen-2.installation.example.net:

co_imc:
  hosts:
    imc-1.installation.example.net:
    imc-2.installation.example.net:
    imc-3.installation.example.net:

co_lb_core_auth:
  hosts:
    auth-1.installation.example.net:
    auth-2.installation.example.net:

co_infra:
  hosts:
    log.installation.example.net:

co_setup:
  hosts:
    auth-1.installation.example.net:
    auth-2.installation.example.net:
    cvm-1.installation.example.net:
    cvm-2.installation.example.net:
    dcm-1.installation.example.net:
    dcm-2.installation.example.net:
    dcm-3.installation.example.net:
    etcd-1.installation.example.net:
    etcd-2.installation.example.net:
    etcd-3.installation.example.net:
    etcd-4.installation.example.net:
    etcd-5.installation.example.net:
    fm-1.installation.example.net:
    fm-2.installation.example.net:
    imc-1.installation.example.net:
    imc-2.installation.example.net:
    imc-3.installation.example.net:
    log.installation.example.net:
    mq-1.installation.example.net:
    mq-2.installation.example.net:
    mq-3.installation.example.net:
    nm-1.installation.example.net:
    nm-2.installation.example.net:
    pregen-1.installation.example.net:
    pregen-2.installation.example.net:
```

ПРИЛОЖЕНИЕ В

Пример файла inventory PGS (установка Standalone)

```

all:
  children:
    pgs:
      children:
        pythagoras:
          hosts:
            host.installation.example.net:
        keycloak:
          hosts:
            host.installation.example.net:
        arangodb:
          hosts:
            host.installation.example.net:
            volume_device_arangodb: "False"
            volume_device_arangodb_path: \
"/dev/disk/by-uuid/<UUID>"
          arangodb_agent:
            hosts:
              host.installation.example.net:
        search:
          hosts:
            host.installation.example.net:
            volume_device_elasticsearch: "False"
            volume_device_elasticsearch_path: \
"/dev/disk/by-uuid/<UUID>"
          redis:
            hosts:
              host.installation.example.net:
        rabbitmq:
          hosts:
            host.installation.example.net:
        postgres:
          hosts:
            host.installation.example.net:
            volume_device_postgres: "False"
            volume_device_postgres_path: \
"/dev/disk/by-uuid/<UUID>"
          etcd:
            hosts:
              host.installation.example.net:
          nginx:
            hosts:
              host.installation.example.net:
          infrastructure:
            hosts:
              host.installation.example.net:
        storage:
          hosts:
            host.installation.example.net:

```

ПРИЛОЖЕНИЕ Г

Пример файла inventory PGS (кластерная установка)

```

all:
  children:
    pgs:
      children:
        pythagoras:
          hosts:
            host1.installation.example.net:
            host2.installation.example.net:
        keycloak:
          hosts:
            host1.installation.example.net:
            host2.installation.example.net:
        arangodb:
          hosts:
            host1.installation.example.net:
              volume_device_arangodb: False
              volume_device_arangodb_path: "/dev/disk/by-uuid/<UUID>"
            host2.installation.example.net:
              volume_device_arangodb: False
              volume_device_arangodb_path: "/dev/disk/by-uuid/<UUID>"
        arangodb_agent:
          hosts:
            host1.installation.example.net:
              volume_device_agent: False
              volume_device_agent_path: "/dev/disk/by-uuid/<UUID>"
            host2.installation.example.net:
              volume_device_agent: False
              volume_device_agent_path: "/dev/disk/by-uuid/<UUID>"
            host3.installation.example.net:
              volume_device_agent: False
              volume_device_agent_path: "/dev/disk/by-uuid/<UUID>"
        search:
          hosts:
            host1.installation.example.net:
              volume_device_elasticsearch: False
              volume_device_elasticsearch_path: \
"/dev/disk/by-uuid/<UUID>"
            host2.installation.example.net:
              volume_device_elasticsearch: False
              volume_device_elasticsearch_path: \
"/dev/disk/by-uuid/<UUID>"
            host3.installation.example.net:
              volume_device_elasticsearch: False
              volume_device_elasticsearch_path: \
"/dev/disk/by-uuid/<UUID>"
        redis:
          hosts:
            host1.installation.example.net:
            host2.installation.example.net:
        rabbitmq:
          hosts:
            host1.installation.example.net:
            host2.installation.example.net:
            host3.installation.example.net:
        etcd:
          hosts:

```



```
    host1.installation.example.net:
    host2.installation.example.net:
    host3.installation.example.net:
  nginx:
    hosts:
      host1.installation.example.net:
      host2.installation.example.net:
  postgres:
    hosts:
      host1.installation.example.net:
        volume_device_postgres: False
        volume_device_postgres_path: "/dev/disk/by-uuid/<UUID>"
      host-2.installation.example.net:
        volume_device_postgres: False
        volume_device_postgres_path: "/dev/disk/by-uuid/<UUID>"
  infrastructure:
    hosts:
      infra.installation.example.net:
  co_lb:
    hosts:
      co-lb1.installation.example.net:
      co-lb2.installation.example.net:
  co_auth:
    hosts:
      co-auth1.installation.example.net:
      co-auth2.installation.example.net:
  storage: #3 minimum
    hosts:
      host1.installation.example.net:
      host2.installation.example.net:
      host2.installation.example.net:
```