ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

СИСТЕМА ХРАНЕНИЯ ДАННЫХ (PGS)

3.3.1

РУКОВОДСТВО ПО УСТАНОВКЕ

Версия 2

На 59 листах

Дата публикации: 17.04.2025

Москва
2025

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

Товарные знаки «МойОфис», «MyOffice» и «Squadus» 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

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

В настоящем документе применяют следующие сокращения с соответствующими расшифровками (табл. 1).

Таблица 1— Сокращения и обозначения

Сокращение, термин	Расшифровка и определение
API	Application Programming Interface, интерфейс программирования приложений
CA	Certificate Authority, удостоверяющий центр для подтверждения подлинности ключей шифрования
CO	CloudOffice, Облачный Офис, общее название продукта (группы редакторов)
DNS	Domain Name System, система доменных имен
ETCD	Распределенная система хранения конфигурации
FQDN	Fully Qualified Domain Name, полностью определенное имя домена
Inventory	Файл для настройки Ansible с перечислением ролей и их IP-адресов
PGS	Pythagoras, сервисы файлового хранилища, работающие по протоколам PGS (Web API, App API, Card API)
SMTP	Simple Mail Transfer Protocol, протокол передачи почтовых сообщений
SSL	Secure Sockets Layer, криптографический протокол
SSH	Secure Shell, «безопасная оболочка»
SSO	Single Sign-On, подсистема единого входа (аутентификации и авторизации)
URL	Uniform Resource Locator, единый указатель ресурса
ОС	Операционная система
ПК	Персональный компьютер
ПО	Программное обеспечение
УЦ	Удостоверяющий центр

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение

Настоящее руководство описывает порядок установки и настройки Системы хранения данных (PGS).

1.2 О компонентах

Система хранения данных — компонент, предназначенный для безопасного хранения корпоративных файлов и обеспечения возможностей авторизации, аутентификации и разграничения прав доступа пользователей.

Система хранения данных входит в состав следующих продуктов:

– «МойОфис Документы Онлайн»;

– «МойОфис Профессиональный 2»;

– «МойОфис Профессиональный 3»;

– «МойОфис Схема»;

– «Squadus PRO».

Список возможностей приложения приведен в документе «Функциональные возможности».

1.3 Перечень изменений текущей версии

Изменения в версии 3.3

1. Добавлена установка и предварительная настройка скриптов резервного копирования в составе дистрибутива.
2. Переименованы общие переменные с CO для упрощения настройки дистрибутивов и сохранена совместимость со старыми файлами inventory.
3. Добавлена возможность обновления прикладных приложений PGS без полной переустановки всего продукта.
4. Изменились в DNS-зоны, необходимые для работы и установки PGS.
5. Добавлен мониторинг истечения сроков жизни паролей для пользователей pgs/app-co (отображение в Grafana)
6. Добавлена возможность экспортировать метрики через open telemetry collector во внешний мониторинг.
7. Добавлена совместимость с Debian 11/12 и Astra 1.8 и подтверждена совместимость с Astra 1.7.6 и Redos 1.8.

1.4 Состав дистрибутива

Дистрибутив PGS представляет собой архив в формате *.tgz и включает в себя:

– набор Ansible плейбуков для развертывания ролей;

– архив образа Docker Registry;

– набор контейнеров для запуска PGS;

– файлы хеша в форматах MD5 и SHA256.

1.5 Перечень технической документации

Перечень технической документации, представленный в таблице 2, предназначен для развертывания серверной части, настройки и дальнейшего администрирования продукта.

Комплект документации распространяется на компоненты продукта:

– Система редактирования и совместной работы (CO);

– Система хранения данных (PGS).

Таблица 2 — Перечень технической документации

Наименование документа	Используемые компоненты	Содержание документа
«Системные требования»	CO, PGS	Системные и программные требования к продукту
«Архитектура»	CO, PGS	Описание архитектуры продукта для выбора типа установки и выделения ресурсов для серверов
«Система редактирования и совместной работы. Руководство по установке»	CO	Порядок установки системы редактирования и совместной работы
«Система хранения данных. Руководство по установке»	PGS	Порядок установки системы хранения данных
«Руководство по настройке»	CO, PGS	Настройка серверов продукта после установки и в ходе эксплуатации системы, а также процессов мониторинга и логирования
«Руководство по администрированию»	CO, PGS	Функции управления тенантом в ходе эксплуатации системы
«Руководство по резервному копированию»	PGS	Порядок резервного копирования баз данных, расположенных в системе хранения данных
«Руководство по мониторингу»	CO, PGS	Функции отображения текущего состояния системы и отдельных сервисов
«Руководство по работе с API»	CO, PGS	Набор методов для автоматизированного управления пользователями, группами, общими папками, доменами и тенантами

1.6 Требования к персоналу

Для работы с ПО Администратору необходимо обладать релевантным опытом по следующим направлениям:

1. Основы сетевого администрирования:
   – сетевая модель OSI и стек протоколов TCP/IP;
   – IP-адресация и маски подсети;
   – маршрутизация: статическая и динамическая;
   – протокол обеспечения отказоустойчивости шлюза (VRRP).
2. Работа с подсистемой виртуализации:
   – работа с VMware vSphere ESXi 6.5 или KVM;
   – установка Docker;
   – запуск, остановка и перезапуск контейнеров;
   – работа с реестром контейнеров;
   – получение параметров контейнеров;
   – взаимодействие приложений в контейнерах (сеть в Docker);
   – решение проблем контейнерной виртуализации.
3. Работа с командной строкой ОС Linux:
   – опыт системного администрирования Linux;
   – знания в объеме курсов AL-1702, AL-1703 (или аналогичных курсов других ОС);
   – знания в объеме, достаточном для сдачи сертификационного экзамена ALCSA-1.7 (или аналогичных экзаменов других ОС).
4. Работа со службой доменных имен DNS:
   – знание основных терминов (DNS, IP-адрес);
   – понимание принципов работы DNS (корневые серверы, TLD-серверы, серверы имен доменов, разрешающий сервер имен);
   – знание типов записи и запросов DNS.

5. Знание видов архитектуры, а также основных компонентов инфраструктуры открытых ключей (PKI);
   - закрытый и открытый ключи;
   - сертификат открытого ключа;
   - регистрационный центр (RA);
   - сертификационный центр (CA);
   - хранилище сертификатов (CR).
6. Работа с системой автоматизации развертывания Ansible.
7. Практический опыт администрирования на уровне эксперта:
   - СУБД ArangoDB;
   - файловой системы GlusterFS;
   - SSO-сервиса Keycloak;
   - СУБД PostgreSQL;
   - поисковой системы Elasticsearch;
   - Redis;
   - обработчика сообщений RabbitMQ;
   - сервера конфигурации ETCD.

1.7 Типовые схемы установки

Структура сервиса может быть представлена двумя типами установки:

- standalone — на один виртуальный сервер или на несколько виртуальных серверов в рамках одного физического сервера;
- кластерная — все роли устанавливаются на разные виртуальные или физические серверы.

1.7.1 Standalone

Конфигурация без отказоустойчивости используется для разработки или демонстрации возможностей продукта.

Для установки продукта в минимальной конфигурации необходимо использовать три сервера:

- сервер с ролью `operator` для управления процессом установки;
- сервер с ролью `CO` для установки редакторов и дополнительного ПО;
- сервер с ролью `PGS` для размещения и хранения базовых библиотек и файлов.

1.7.2 Кластерная установка

Кластерная установка — отказоустойчивая конфигурация, используемая для типовой установки продукта.

Для сохранения уровня отказоустойчивости не рекомендуется совмещать серверные роли между собой. Совмещение допускается в отдельных случаях для экономии ресурсов.

Для минимальной конфигурации кластерной установки необходимо использовать не менее трех серверов. Примеры конфигурации представлены в дистрибутиве продукта `~/install_MyOffice_PGS/inventory/`

1.8 Порядок установки серверов

1. Необходимо подготовить сервер с ролью `operator` в соответствии с разделом «Подготовка сервера с ролью operator».

   В качестве сервера с ролью operator может использоваться рабочий компьютер пользователя, отвечающий требованиям, указанным в документе «Системные требования».
2. Если комплект поставляемого ПО включает в себя продукт «МойОфис Почта 3», то в первую очередь необходимо выполнить установку почтового сервера с помощью сервера с ролью `operator`. Порядок установки почтового сервера представлен в документе «Руководство по установке почтового сервера».
3. С помощью сервера с ролью `operator` необходимо подготовить инфраструктуру и выполнить установку Системы хранения данных в соответствии с настоящим руководством.
4. С помощью сервера с ролью `operator` следует подготовить инфраструктуру и выполнить установку Системы редактирования и совместной работы (СО) в соответствии с документом «Система редактирования и совместной работы (СО). Руководство по установке».
5. После установки продукта сервер с ролью `operator` используется для переустановки отдельных сервисов, замены SSL-сертификатов, переустановки или обновления продукта. Сервер хранит настроенные конфигурационные файлы, содержащие пароли от сервисов. Необходимо ограничить доступ к серверу с ролью operator в целях обеспечения безопасности.
6. С помощью документов по настройке, перечисленных в разделе «Перечень технической документации», выполнить необходимые интеграции и установить параметры сервисов.

1.9 Программные и аппаратные требования

Программные и аппаратные требования к текущей версии ПО указаны в документе «Системные требования».

2 ПОДГОТОВКА К УСТАНОВКЕ

2.1 Подготовка серверов установки

Перед началом установки необходимо ознакомиться с документом «Архитектура». В соответствии с типом установки следует подготовить необходимое количество физических или виртуальных серверов.

2.2 Подготовка ОС

На серверы, предназначенные для развертывания системы, необходимо установить ОС, соответствующую требованиям документа «Системные требования».

Для установки на ОС Astra необходимо выполнить операции, изложенные в разделе «Конфигурирование ОС Astra».

2.2.1 Конфигурирование ОС Astra

2.2.1.1 Установка на Astra SE 1.7 в защищенных вариантах

Основные отличия между вариантами защищенности Astra SE 1.7 приведены в таблице 3.

Таблица 3 — Уровни защищенности ОС Astra

Функция безопасности	Уровень защиты «Базовый»*	Уровень защиты «Усиленный»*	Уровень защиты «Максимальный»*
Замкнутая программная среда	Не доступна	Доступна (по умолчанию выключена)	Доступна (по умолчанию выключена)
Очистка освобождаемой внешней памяти	Не доступна	Доступна (по умолчанию выключена)	Доступна (по умолчанию выключена)
Мандатный контроль целостности	Не доступна	Доступна (по умолчанию включена)	Доступна (по умолчанию включена)
Мандатное управление доступом	Не доступна	Не доступна	Доступна (по умолчанию включена)
* — наименование ОС Астра в соответствии с уровнем защиты: – Базовый уровень — Астра 1.7 «Орел»; – Усиленный уровень — Астра 1.7 «Воронеж»; – Максимальный уровень — Астра 1.7 «Смоленск»			

Текущий уровнень защищенности проверяется с помощью команды:

```
root@voronezh:~# astra-modeswitch list
0   base(orel)
1   advanced(voronezh)
2   maximum(smolensk)
root@voronezh:~# astra-modeswitch get
1
```

Текущий статус замкнутой программной среды проверяется с помощью команды:

```
root@voronezh:~# astra-digsig-control status
ACTIVE
```

Текущий статус очистки освобождаемой внешней памяти (очистка разделов подкачки и гарантированное удаление файлов) проверяется с помощью команды:

```
root@voronezh:~# astra-swapwiper-control status
ACTIVE
root@voronezh:~# astra-secdel-control status
ACTIVE
on /
```

Текущий статус мандатного контроля целостности проверяется с помощью команды:

```
root@voronezh:~# astra-mic-control status
ACTIVE
```

Текущий статус мандатного управления доступом проверяется с помощью команды:

```
root@voronezh:~# astra-mac-control status
INACTIVE
```

Текущий статус запрета включения бита выполнения проверяется с помощью команды:

```
root@voronezh:~# astra-nochmodx-lock status
ACTIVE
```

2.2.1.2 Установка на усиленном уровне защищенности («Воронеж»)

Установка осуществляется Ansible от имени пользователя astra, для которого должна быть настроена возможность выполнять sudo без пароля.

1. Пользователю astra необходимо установить максимальный уровень целостности — 63 (соответствует администратору ОС). Для проверки уровня целостности пользователя необходимо выполнить следующую команду:

```
root@voronezh:~# pdp-id -i
63
```

2. Установка Ansible и работа невозможна при включенном запрете бита исполнения. Перед началом установки на всех серверах следует выполнить команды:

```
astra@voronezh:~$ sudo astra-nochmodx-lock disable
astra@voronezh:~$ sudo astra-nochmodx-lock status
INACTIVE
```


3. Установка Ansible и работа PGS невозможна при включенном режиме замкнутой программной среды. Для проверки статуса режима необходимо выполнить следующую команду:

```
astra@voronezh:~$ sudo astra-digsig-control status
INACTIVE
```

4. При статусе ACTIVE перед началом установки на всех серверах следует выполнить команды:

```
astra@voronezh:~$ sudo astra-digsig-control disable
astra@voronezh:~$ sudo reboot
astra@voronezh:~$ sudo astra-digsig-control status
INACTIVE
```

5. Необходимо проверить статусы параметров безопасности, значения которых должны соответствовать таблице 4.

Таблица 4 — Параметры безопасности по умолчанию

Наименование команды	Статус
astra-bash-lock status	INACTIVE
astra-commands-lock status	INACTIVE
astra-docker-isolation status	INACTIVE
astra-hardened-control status	INACTIVE
astra-interpreters-lock status	ACTIVE
astra-lkrg-control status	INACTIVE
astra-macros-lock status	INACTIVE
astra-modban-lock status	INACTIVE
astra-overlay status	INACTIVE
astra-ptrace-lock status	ACTIVE
astra-sumac-lock status	INACTIVE
astra-shutdown-lock status	INACTIVE
astra-ufw-control status	INACTIVE
astra-ulimits-control status	INACTIVE

6. Для проверки доступности репозиториев необходимо выполнить команду:

```
apt-get update
```

Команда должна завершаться без ошибки.

При наличии сбойного зеркала репозитория (например, http://mirror.yandex.ru/astra/stable/orel/repository orel InRelease), его необходимо удалить из директории `/etc/apt/sources.list`.

2.2.2 Порядок обновления ядра Linux

При установке ОС на серверы кластера ядро может быть автоматически обновлено до минимальной требуемой версии.

Настройки по умолчанию для ОС:

– в ОС Ubuntu и РЕД ОС по умолчанию ядро обновляется;

– в ОС Альт автоматическое обновление ядра не поддерживается.

2.3 Настройка сетевых соединений

Настройку сетевого соединения необходимо выполнить на всех серверах, предназначенных для установки системы, в том числе на сервере с ролью `operator`.

Для работы серверов в локальной сети необходимо задать следующие параметры:

– IP-адрес;

– маска подсети;

– основной шлюз;

– DNS-сервер.

2.4 Повышение отказоустойчивости при настройке DNS

Для повышения отказоустойчивости на всех серверах системы рекомендуется указать два или более DNS-сервера резолвера с использованием следующих опций:

– `attempts` — число попыток отправки запроса к серверу;

– `timeout` — время ответа от сервера (по умолчанию — 5 сек.);

– `rotate` — случайный выбор сервера nameserver из списка (по умолчанию — опрос по порядку).

Пример записи в файле `etc/resolv.conf`

```
nameserver 8.8.8.8
nameserver 8.8.4.4
options timeout:2 attempts:2 rotate
```


2.5 Подготовка сервера с ролью operator

2.5.1 Установка дополнительного ПО

В соответствии с документом «Системные требования» на сервере с ролью `operator` необходимо установить пакеты дополнительного ПО.

Рекомендуется использовать «чистую» ОС для предотвращения появления ошибок, связанных с использованием конфигурационных файлов.

Конфигурационные файлы, установленные по умолчанию (например: `/etc/ansible/ansible.cfg`), необходимо удалить или заменить файлами из комплекта поставляемого ПО.

Для установки пакетов необходимо обеспечить серверу с ролью `operator` выход в интернет.

Запрещается установка последних версий дополнительного ПО, доступных в репозитории. Перед установкой следует ознакомиться с требованиями к версиям ansible-core и модулям Python.

2.5.2 Установка в сети без выхода в интернет

Для установки продукта в локальной сети, без прямого выхода в интернет, необходимо обеспечить доступность дополнительных пакетов ПО. Перечень необходимого ПО приведен в документе «Системные требования».

Для обеспечения доступности необходимых для установки пакетов следует:

1. Скачать пакеты и их зависимости со стандартного репозитория с помощью прямого выхода в интернет, выполнив команду:

```
python3 -m pip download "bcrypt==3.1.7" "docker==6.0.0" "passlib" "pyyaml" \
"jsondiff" "requests<2.29.0" "pip<21.0" "urllib3<2.0.0" \
"websocket-client==0.59.0"
```

Модули Python, устанавливаемые из pip или системных пакетов дополнительно представлены в дистрибутиве PGS в директории offline_install

2. Перенести скачанные пакеты и их зависимости на все сервера установки. Пример копирования файлов:

```
scp <file> <remote_user>@<remote_server>:/<path_on_remote_server>
```

где:

`<file>` — имя скачанного пакета;

`<remote_user>` — имя пользователя;

`<remote_server>` — IP-адрес или DNS-имя сервера из группы серверов установки, представленных в файле `hosts.yml`;

`<path_on_remore_server>` — каталог размещения пакетов.

3. Выполнить установку скачанных пакетов с помощью команды:

```
pip install --no-index --find-links ./<file>
```

где:

`<file>` — имя скачанного пакета;

2.6 Подготовка инфраструктуры установки

2.6.1 Проверка и подготовка дистрибутива ПО

Для подготовки и проверки дистрибутива необходимо:

1. После копирования архива проверить его контрольную сумму и сравнить значение с данными, полученными от вендора ПО:

– для MD5 с помощью команды:

```
md5sum -c MyOffice_PGS_3.3.1.tar.gz.md5
```

– для SHA256 с помощью команды:

```
sha256sum -c MyOffice_PGS_3.3.1.tar.gz.sha256
```

2. Распаковать содержимое архива в произвольный каталог и перейти в него:

```
mkdir install_MyOffice_PGS
tar xf MyOffice_PGS_3.3.1.tgz -C install_MyOffice_PGS
cd install_MyOffice_PGS
```

Не рекомендуется распаковывать новый дистрибутив в каталог предыдущей версии.

2.6.2 Настройка DNS

Перед началом установки необходимо настроить DNS-сервер, указав адрес установки сервера Nginx (см. таблицу 5).

Таблица 5 — Настройка DNS

Доменное имя с использованием <env>	**Доменное имя без использования** <env>	**Хост**	**Описание**
internal-<env>.<default_domain>	internal.<default_domain>	Nginx host	Адрес веб-панели администрирования PGS
pgs-<env>.<default_domain>	pgs.<default_domain>	Nginx host	Адрес точки входа для API

Необходимо использовать одинаковый домен для устанавливаемых компонентов продукта (см. раздел Общие переменные для CO и PGS). Запрещается использовать в качестве домена зону *.local

Переменные `<env>` и `<default_domain>` заполняются в соответствии с разделом «Конфигурирование переменных файла hosts.yml» данного руководства. Nginx host соответствует адресу, указанному в файле inventory для роли `nginx` (подробнее см. в разделе «Конфигурирование ролей файла hosts.yml»).

Если переменная `<env>` не задана, то запись будет `internal.<default_domain>`

Если переменная `<env>` задана, то запись будет `internal-<env>.<default_domain>`

При обновлении системы на версию 3.3 необходимо выполнить требования данного раздела по измению DNS-записи

При обновлении системы на версию 3.3 следует изменить DNS-запись `admin-<env>.<default_domain>` (`admin.<default_domain>`) на DNS-запись `internal-<env>.<default_domain>` (`internal.<default_domain>`).

Настройка внешнего доступа к DNS-записи `admin-<env>.<default_domain>` (`admin.<default_domain>`) выполняется при установке Системы редактирования и совместной работы.

2.6.3 Настройка сертификатов

Для работы веб-интерфейса PGS необходима установка SSL-сертификатов. Сертификаты необходимо разместить в каталоге, соответствующему доменному имени PGS (`<default_domain>`).

Пример расположения каталога:

```
~/install_MyOffice_PGS/certificates/<default_domain>
```

где `~/install_MyOffice_PGS` — корневой каталог установки.

Подробное описание переменных см. в разделе «Конфигурирование переменных файла hosts.yml» данного руководства.

Список необходимых сертификатов размещен в таблице 6.

Таблица 6 — Перечень необходимых сертификатов

Наименование сертификата	Описание
server.crt	Содержит SSL-сертификат для *.<default_domain> и все промежуточные сертификаты, кроме корневого доверенного. Расположение промежуточных сертификатов соответствует описанию в документации Nginx
server.nopass.key	Приватный ключ сертификата, не требующий кодовой фразы
ca.crt	При наличии самоподписанных или не публичных доверенных SSL-сертификатов

Рекомендуется использовать сертификаты, полученные от публичных центров сертификации.

2.6.4 Создание самоподписанного сертификата

Для создания самоподписанного сертификата в PGS необходимо запустить исполняемый файл `gen_self_signed_cert.sh` из каталога установки. При запуске файла указывается домен, привязанный к создаваемому сертификату.

Пример команды создания сертификата:

```
bash gen_self_signed_cert.sh <DOMAIN>
```

После создания файл сертификата будет автоматически размещен в необходимом каталоге (см. в разделе «Настройка сертификатов»).

2.7 Настройка параметров установки

Директория установки содержит предзаполненные файлы конфигураций, подготовленные для упрощения настройки системы. Необходимо скопировать шаблон файла inventory в корневой каталог дистрибутива и заполнить секции `hosts` и `vars`. Шаблоны для заполнения в зависимости от типа конфигурации представлены в таблице 7.

Таблица 7 — Шаблоны файлов конфигурации

Тип конфигурации	Расположение шаблона
Конфигурация без отказоустойчивости	~/install_MyOffice_PGS/inventory/hosts-sa.yaml
Кластерная установка	~/install_MyOffice_PGS/inventory/hosts-hl.yaml

Файл inventory использует формат .yaml, синтаксис которого описан в документации Ansible.

Операция копирования выполняется с помощью команды:

```
cp ~/install_MyOffice_PGS/inventory/hosts-sa.yaml hosts.yml
```

После заполнения файл конфигурации рекомендуется хранить отдельно на внешнем ресурсе. Файл может потребоваться при восстановлении и/или переустановке системы.

2.7.1 Конфигурирование ролей файла hosts.yml

Для определения роли сервера необходимо добавить его доменное имя или IP-адрес в соответствующую секцию в шаблоне файла inventory. После назначения роли серверу при установке будут выполнены команды Ansible.

Домен host.example.com (указан для примера) должен резолвится в IP-адрес на сервере с ролью operator и на всех серверах кластера.

Пример. Для назначения группы ролей `pythagoras` серверу с доменным именем `host.example.com` необходимо указать следующие значения:

```
pythagoras:
  hosts:
    host1.example.com:
```

При совмещении всех ролей на одном сервере в шаблоне файла inventory дублируется секция `hosts`. При изменении конфигурации установки возможно добавление или удаление серверов в группах.

В режиме кластерной установки в файле inventory указывается несколько IP-адресов или доменных имен серверов в соответствующей группе.

Пример (фрагмент шаблона `hosts-sa.yaml`). Все роли и группы ролей устанавливаются на один сервер по адресу `host.example.com`:

```
all:
  children:
    pgs:
      children:
        pythagoras:
          hosts:
            host.example.com:
        keycloak:
          hosts:
            host.example.com:
        arangodb:
          hosts:
            host.example.com:
              volume_device_arangodb: False
              volume_device_arangodb_path: "/dev/disk/by-uuid/<UUID>"
```

Текущей версией ПО поддерживается кластеризация для сервисов, перечисленных в таблице 8. В таблице указано минимально необходимое количество серверов для работы кластера. В зависимости от инфраструктуры и типа установки количество серверов может быть изменено.

Таблица 8 — Поддержка кластеризации

Наименование сервиса	Группа	Количество серверов
Aristoteles Dionis Epicure Euclid Heraclitus Pheidippides Polemon Restrictions Sisyphusworker Users User_listener	Pythagoras	2
		1
Keycloak	Keycloak	2
ArangoDB	ArangoDB	2(1)
	Arangodb_agent	3(1)
Elasticsearch	Search	3
Sisyphussearch		
Redis	Redis	3
RabbitMQ	RabbitMQ	3
Etcd	Etcd	3
Nginx	Nginx	2
Postgres	Postgres	2
Docker Registry	Infrastructure	1
Syslog-ng		
Monitoring		
Storage	Storage	3

2.7.1.1 Конфигурация для кластерной установки

Пример конфигурации для кластерной установки находится в шаблоне `hosts-hl.yaml`. Группа хостов `arangodb_agent` используется для кластерной установки с использованием `agent`.

Для работы группы необходимо выделить не менее 3-х отдельных хостов (количество хостов должно быть нечетным числом). В ином случае группу следует оставить незаполненной:

```
arangodb_agent:
        hosts:
```

Роли `arangodb`, `arangodb_agent`, `search`, `postgres` содержат дополнительные переменные `volume_device_<role>` и `volume_device_<role>_path`, необходимые для хранения данных на блочных устройствах, форматированных в файловую систему XFS.

Пример значений для переменных:

```
volume_device_<role>: "True"
      volume_device_<role>_path: "<filesystem_path>"
```

Где `<role>` — логическая роль, `<filesystem_path>` — путь до файловой системы устройства.

Особенности работы в режиме `volume_device_<role>: "True"`:

1. Не допускается использование одного и того же раздела диска на одном сервере (или виртуальной машине) для нескольких ролей.

2. Диск следует отформатировать в файловую систему XFS. На момент развертывания системы диск должен быть размонтирован (кроме ситуации повторного запуска).

В режиме `volume_device_<role>: "False"` действий от пользователя не требуется, данные хранятся в соответствующих каталогах:

```
/var/lib/docker/volumes/<volume_name>
```

Где `<volume_name>` — том (каталог Docker), привязанный к контейнеру устанавливаемой роли.

Допускается использование для некоторых ролей режима `volume_device_<role>: "True"`, а для других `volume_device_<role>: "False"`.

При кластерной установке продукта потребуется настройка балансировщика нагрузки между PGS и его auth-нодами. Для этого в inventory файле PGS предусмотрены две группы:

- `co_lb` — группа хостов, на которых будет установлен и настроен сервис балансировки нагрузки `keepalived`;
- `co_auth` — группа, в которой нужно указать сетевые адреса auth-нод CO.

Дополнительная информация по интеграции с CO описана в документе «Руководство по настройке».

2.7.1.2 Сбор событий и метрик, хранение образов

Группа ролей `infrastructure` служит для хранения образов установки, а также сбора событий и метрик мониторинга системы. Их работа не блокирует работу PGS.

События, собираемые со всех серверов установки сервисом Syslog-ng, будут храниться на сервере, назначенном группе ролей `infrastructure` в файле inventory. Путь к журналу событий будет выглядеть следующим образом:

```
/var/log/pgs/<env>.<default_domain>/<service_name>/<element>.log
```

Где:

- `<env>.<default_domain>` — переменные, заполненные в соответствии с разделом «Конфигурирование переменных файла hosts.yml»;
- `<service_name>` — имя сервиса;
- `<element>` — название файла лога.

2.7.2 Конфигурирование переменных файла hosts.yml

Процесс настройки переменных файла inventory состоит в заполнении секции `vars`. Доступные значения и способы заполнения секции указаны в таблице 9.

Параметры переменных необходимо указывать в двойных кавычках. Спецсимволы «<>{}|&;*?@`$!» в значениях переменных необходимо экранировать символом «\». Для обеспечения безопасности при работе ПО рекомендовано использовать надежные пароли, содержащие спецсимволы и произвольные символы разных регистров.

Доступ к сервисам PGS обеспечивается с помощью переменных:

- `DEV_MODE`;
- `PGS_CLUSTER`;
- `DEFAULT_DOMAIN`;
- `ENV`;
- `SWARM_NETWORK_ENCRYPTION`;
- `API_INTERFACE_EXT_PORT`.

После заполнения перечисленных переменных будет сформирован адрес: `https://pgs-<env>.<default_domain>:<admin_interface_ext_port>`, который служит для обеспечения доступа к сервису. Если переменная `<env>` не задана, то запись будет `https://pgs.<default_domain>:<admin_interface_ext_port>`

Таблица 9 — Переменные секции vars

Переменная	Значение и способ заполнения
PGS_CLUSTER	Включение и отключение кластерного режима установки системы. Принимает значения True и False. В шаблоне hosts-sa.yaml по умолчанию — False, в шаблоне hosts-hl.yaml по умолчанию — True, в шаблоне hosts-hl-sa.yaml по умолчанию — True
ARANGO_CLUSTER	Включение и отключение кластерного режима установки сервиса Arango. Принимает значения True и False. В шаблоне hosts-sa.yaml и hosts-hl-sa.yaml по умолчанию — False, в шаблоне hosts-hl.yaml по умолчанию — True
DEFAULT_DOMAIN	Зарегистрированный домен установки PGS. Для корректной работы необходим установленный актуальный SSL-сертификат
ENV	Элемент доменного имени установки, предназначеный для разграничения доступа к сервисам PGS
ADMIN_INTERFACE_ EXT_PORT	Порт Nginx для доступа к интерфейсу администратора, значение по умолчанию — 443. При изменении значения по умолчанию для порта в PGS необходимо учесть новое значение в CO (Подробнее см. в документе «Система редактирования и совместной работы (CO). Руководство по установке»).

Переменная	Значение и способ заполнения
	При включении интеграции с почтовой системой PSN следует изменить порт для доступа к интерфейсу администратора PGS в PSN (Подробнее см. в документе «Руководство по установке почтового сервера»)
API_INTERFACE_ EXT_PORT	Порт Nginx для доступа к API интерфейсу, значение по умолчанию — 443. При изменении значения по умолчанию для порта в PGS необходимо учесть новое значение в компоненте CO (Подробнее см. в документе «Система редактирования и совместной работы (CO). Руководство по установке»). При включении интеграции с почтовой системой PSN следует изменить порт для доступа к pgs_api в параметрах PSN (Подробнее см. в документе «Руководство по установке почтового сервера»)
CUSTOM_CA	Заполняется при использовании самоподписанных сертификатов, допустимые значения: True или False. При значении True — файл ключа (например, в формате .crt) размещается в директории Certificates в корневом каталоге установки
HERACLITUS_CRON	Задает время запуска сервиса Heraclitus. Для определения времени запуска используется формат, аналогичный формату Cron. Значение по умолчанию — "0 2 * * *".
KEYCLOAK_PASSWORD	Пароль для пользователя PGS в Keycloak (он же Администратор Master Realm). Длина значения не менее 10 символов
KEYCLOAK_REALM_ PASSWORD	Внутренний пароль для администраторов тенантов Keycloak (используется только для сервисного обслуживания системы). Длина значения не менее 10 символов
KEYCLOAK_POSTGRES_ PASSWORD	Пароль БД PostgreSQL (используется как хранилище для Keycloak). Длина значения не менее 10 символов
ARANGODB_ PASSWORD	Пароль пользователя PGS в ArangoDB
pgs_rabbitmq_password	Пароль пользователя RabbitMQ
REDIS_PASSWORD	Пароль доступа в Redis
PATRONI_REPLICATION_P ASSWORD	Пароль для репликации БД PostgreSQL (только для кластерной установки)
GRAFANA_ADMIN_ PASSWORD	Пароль доступа к интерфейсу Grafana в случае установки с ключом: `-e monitoring_enable=true`
SELINUX_ENABLED	Проверяет режим работы SELinux и переключает его в режим enforcing. Используется только для RedHat-based ОС. Доступные значения: True и False, по умолчанию — False
Блок default_tenant	
Блок default_tenant	Предназначен для создания тенанта по умолчанию, необходимого для дальнейшей работы с пользователями в веб-интерфейсе PGS

Переменная	Значение и способ заполнения
ADMIN_PASSWORD	Пароль для администрирования тенанта (обязательный параметр, при отсутствии значения тенант создан не будет)
ADMIN_RECOVERY_EMAIL	Почта для восстановления доступа к тенанту (обязательный параметр, при отсутствии значения тенант создан не будет)
MAX_USERS	Количество пользователей в тенанте, значение по умолчанию — 1000
QUOTA_PER_USER	Выделенное пользователю место в хранилище, указывается в байтах, значение по умолчанию — 1000000000 (~1 Гбайт)
QUOTA_PER_GROUP	Выделенное место для хранения общих папок. указывается в байтах, значение по умолчанию — 10737418240 (10 Гбайт)
Блок storage	
Блок storage	Блок настроек системы хранения файлов
type	Выбор типа системы хранения файлов, доступны значения fs (файловая система) и s3 (объектное хранилище). В качестве хранилища fs в проекте используется GlusterFS. В качестве хранилища s3 в проекте используется MinIO.
Блок fs	
path	Путь до файловой системы хранения fs. Значение должно заканчиваться символом «/». Значение по умолчанию — `/media/storage/`
retention_file_time	Время хранения файлов после удаления из корзины, указывается в днях, значение по умолчанию — 30
Блок s3	
Блок s3	Параметры доступа к хранилищу s3, если используется `storage: type: "s3"`. Информацию по заполнению переменных следует запросить у хостинг-провайдера, ниже приведены указания для заполнения при использовании сервиса MinIO
minio_used	True — если используется сервис MinIO. False — если используется стороннее s3-хранилище
minio_access_key	Переменная задается при использовании сервиса MinIO. Значение в произвольном виде, минимальная длина — 8 символов
minio_secret_key	Переменная задается при использовании сервиса MinIO. Значение в произвольном виде, минимальная длина — 8 символов
use_old_minio	Принимает значения: True/False При значении False из дистрибутива будет установлен MinIO версии RELEASE.2023-12-13T23-28-55Z При значении True будет установлен MinIO максимально совместимой версии RELEASE.2022-06-25T15-50-16Z Переменная со значением False предназначена: 1. Для первичной установки продукта.

Переменная	Значение и способ заполнения
	2. Если в инфраструктуре не используется собственное s3-хранилище. 3. Если в инфраструктуре не предустановлен MinIO ранних версий датой выпуска до 2022-06-26. Переменная со значением true предназначена: 1. Для последующей установки продукта (не первичной). 2. Для использования ранних версий MinIO, при условии что MinIO ранее был установлен (текущая используемая версия выпущена до 2022-06-26). 3. Для обновления с версий 2.7 до актуальной версии продукта, где в качестве storage type использовалось значение s3. Для обновления MinIO с версии RELEASE.2022-06-25T15-50-16Z до RELEASE.2023-12-13T23-28-55Z необходимо использовать официальное руководство https://min.io/docs/minio/linux/operations/install-deploy-manage/migrate-fs-gateway.html
url	Ссылка для доступа к сетевому хранилищу. При использовании MinIO имеет следующий вид: `http://pgs-<ENV>.<DEFAULT_DOMAIN>:9002` Значения `<env>` и `<default_domain>` соответствуют переменным, указанным в начале данной таблицы
secret_key	Параметр, соответствующий настройкам хранилища s3. При использовании сервиса MinIO совпадает со значением переменной `minio_secret_key`
access_key	Параметр, соответствующий настройкам хранилища s3. При использовании сервиса MinIO совпадает со значением переменной `minio_access_key`
bucket	Контейнер для хранения объектов в хранилище s3. При использовании хранилища s3 необходимо указать значение. При отсутствии контейнера — он будет создан во время установки продукта
service_name	При использовании сервиса MinIO указывается значение `s3`
region_name	При использовании сервиса MinIO указывается значение `myoffice`
acl	Сущность для разграничения прав доступа, в случае с MinIO необязательна к заполнению
s3_max_capacity	Параметр, определяющий максимальное пространство, доступное в s3, указывается в байтах
Блок system	
TIMEZONE	Временная зона (часовой пояс) установки в формате базы tz. Значение по умолчанию — `"Europe/Moscow"`
Блок co	
Блок co	Переменные, которые необходимо заполнить для интеграции с CO.

Переменная	Значение и способ заполнения
	Более подробно о заполнении блока см. в документе Система редактирования и совместной работы (CO). Руководство по установке»
coapiurl	Путь доступа к API CO. Данная переменная представляет собой URL-адрес и порт, указывающие на целевой сервер с ролью auth CO. Пример: `coapiurl: "https://co-api-url.ru:8443"`
co_lb	Включает и выключает настройку балансировки с помощью сервиса Keepalived. Принимает значения True и False (только для кластерной установки) При значении True необходимо добавить в inventory файл следующую переменную: `CENTOS7: false`
vip_auth	Виртуальный IP-адрес, доступное значение — произвольный свободный IP-адрес в сети установки (только для кластерной установки)
lb_keepalived_pass	Пароль для сервиса keepalived (только для кластерной установки)

nginx_whitelist_network	Ограниченный доступ к Nginx PGS со стороны CO. В качестве значения переменной необходимо указать IP-адрес или подсеть кластера развертывания компонента CO.
	Примеры заполнения:
	– для одного сервера:
	`nginx_whitelist_network: ["192.168.0.1"]`
	– для подсети:
	`nginx_whitelist_network: ["192.168.0.1/24"]`
	– для группы серверов:
	`nginx_whitelist_network: ["192.168.0.1", "192.168.0.2"]`

Блок installation_commons	
Блок installation_commons	Значения переменных данного блока должны соответствовать аналогичным в CO, за исключением `fs_admin_password`. Значение этой переменной генерируется администратором при установке PGS. Более подробно о заполнении блока см. в документе «Система редактирования и совместной работы (CO). Руководство по установке»
fs_token_salt_ext	
fs_app_encryption_key	
fs_app_encryption_iv	
fs_app_encryption_salt	
auth_encryption_key	
auth_encryption_iv	
auth_encryption_salt	

Переменная	Значение и способ заполнения
fs_admin_password	
openresty_api_username	Имя пользователя для API-авторизации в СО. Должно совпадать со значением переменной `openresty_api_username` в конфигурации СО
openresty_api_password	Пароль для API-авторизации в СО. Должен совпадать со значением переменной `openresty_api_password` в конфигурации СО
TWO_FA_ENCRYPTION_ KEY	Ключ для шифрования 2FA секретов, длиной 64 символа
AUDIT_LOG_ENABLED	Переменная предоставляет возможность включения в административном интерфейсе расширенного лога событий. Доступные значения: `True` и `False`, по умолчанию — `False`
CHATBOT_ENABLED	Включение и отключение интеграции СО с сервисом ChatBot. Значение зависит от наличия сервиса в СО. Доступные значения: `True` и `False`, по умолчанию — `False`
POSEIDON_ INTEGRATION	Включение и выключение интеграции с почтовым сервером, доступные значения: `True` и `False`
Блок POSEIDON	
Блок POSEIDON	Параметры подключения к почтовому серверу, если включена интеграция. Подробные сведения об установке и настройке в документе «Руководство по установке почтового сервера»
PBM_URL	Ссылка для доступа к почтовому серверу в формате https://pbm.example.com
PBM_USER_ PASSWORD	Переменная для авторизации через API . Соответствует значению переменной ds389_manager_user при установке почтового сервера
SSL_VERIFY	Параметры шифрования для почты. Принимает значения True и False, False — в случае использования самоподписанных сертификатов

Переменные, указанные в таблице 10, настроены по умолчанию и определяются в процессе установки системы. Не рекомендуется изменять значения переменных без необходимости.

Для изменения значений по умолчанию дополнительным переменным следует открыть файл `~/install_MyOffice_PGS/hosts.yml` и добавить необходимые переменные в блок vars.

Таблица 10 — Дополнительные переменные

Переменная	Значение и способ заполнения
DEV_MODE	Developers mode, режим разработчика. Принимает значения True и False. При значении True — открывает порты сервисов для внешнего подключения, в целях организации доступа разработчиков к стенду установки (не используется в работающей с пользователями системе)
MAX_TENANTS	Задает максимально возможное число тенантов в текущей установке (максимально допустимое значение 100)
IPTABLES_ENABLED	Устанавливает и настраивает службы межсетевого экрана (опционально). Доступные значения: True и False, по умолчанию — False
ENABLE_CUSTOM_CA_ CONNECTION_S3	Для использования CA сертификатов с хранилищем S3 следует установить значение True, скопировать корневой сертификат в директорию с сертификатами в файл s3_ca.crt Если переменная отсутствует в файле hosts.yaml, значение по умолчанию — False
ENABLE_CUSTOM_CA_ CONNECTION_LDAP	Для использования CA сертификатов с LDAP-сервером и хранилищем S3 следует установить значение True, скопировать корневой сертификат в директорию с сертификатами в файл ldap_ca.crt Если переменная отсутствует в файле hosts.yaml, значение по умолчанию — False
SWARM_NETWORK_ ENCRYPTION	Включает шифрование внутренней оверлейной сети Docker swarm, значение по умолчанию — False. Влияет на производительность системы
SWARM_NETWORK_ ADDRESS	Адресация внутреней сети "pgs-network" в docker swarm, значение по умолчанию "10.0.0.0/24". Используется при пересечении адресации docker swarm с адресацией внутренней сети мест установки
SWARM_NETWORK_GW	Шлюз по умолчанию для внутреней сети "pgs-network", значение по умолчанию "10.0.0.1". Используется при пересечении адресации docker swarm с адресацией внутренней сети мест установки
SWARM_NETWORK_ INGRESS	Адресация ingress сети в docker swarm, значение по умолчанию, "10.0.1.0/24". Используется при пересечении адресации docker swarm с адресацией внутренней сети мест установки
minio_drives_per_node	Определяет количество дисков для каждого сервера в кластере MinIO. Используется при установке PGS совместно с MinIO Значение по умолчанию: 2 (переменная принимает только целочисленные значения)
data_path_port	Изменяет порт передачи данных в docker swarm. Значение по умолчанию 9999

Переменная	Значение и способ заполнения
pgs_observability_mode	Отвечает за выбор систем мониторинга, устанавливаемых при развертывании продукта. Переменная принимает следующие значения: external — устанавливаются только The OpenTelemetry Collector. Для корректной работы необходимо указать значение переменной `otel_collector_external_otlp_endpoint` internal — устанавливаются только сервисы Prometheus и Grafana (используется по умолчанию) both — устанавливаются Prometheus, Grafana и The OpenTelemetry Collector. Для корректной работы необходимо указать значение переменной `otel_collector_external_otlp_endpoint`
otel_collector_external_otlp_endpoint	Определяет IP или доменное имя внешнего хоста, которое принимает данные в формате open telemetry. Значение переменной записывается в формате: `<ip/host name>:<port>`
log_rotate_days	Определяет количество дней хранения логов всех сервисов системы. Если переменная не задана в конфигурационном файле, то ее значение будет установлено в процессе развертывания, 14 дней. Переменная использует целочисленный формат
squadus_integration	Включает интеграцию с сервисом Squadus. Принимает значения True и False. По умолчанию значение False
Настройка подключения к DLP системе	
common_dps_enabled	Включает настройку подключения к сервису DPS при запуске или обновлении продукта. Принимает значения True и False. По умолчанию False.
dps_icap_server_address	Определяет IP-адрес или DNS-имя DPS-сервера компонента CO. Значение переменной необходимо указывать с протоколом, например: https://co-host.example.net
dps_service_port	Определяет порт DPS-сервера компонента CO. По умолчанию значение 20003
dps_service_path	Значение по умолчанию '/api/v1/dlp'
Настройка подключения к SSO	
redis_host	Значение необходимо взять из файла CO inventory из группы хостов co_imc. Для standalone установки переменная принимает тип string `redis_host: "co-infra-1.installation.example.net"` Для кластерной установки переменная принимает тип array

Переменная	Значение и способ заполнения
	Пример значения: `redis_host: "co-infra-1.installation.example.net;co-infra-2.installation.example.net;co-infra-3.installation.example.net"` Перечисляемые хосты в значении должны быть через знак ; без пробелов
redis_password	Пароль для подключения к сервису Redis на стороне CO. Значение переменной должно совпадать с аналогичной переменной в CO
redis_sentinel_port	Порт для подключения к сервису Redis на стороне CO. Значение переменной должно совпадать с аналогичной переменной в CO
common_redis_sentinel_master_name	Имя мастер-узла в кластере Redis. Значение по умолчанию 6379, должно совпадать с аналогичной переменной в CO
Настройка синхронизации статусов пользователей с внешними каталогами	
ldap_required	Значение по умолчанию true добавляет сервис user_listener в систему. Значение false используется для системы без каталога
mapper_set	Значение по умолчанию false.Следует установить true после настройки атрибута в Keycloak. Если маппер не настроен, число реплик при установке равно 0

2.7.3 Рекомендации по настройке дисков для ролей

1. Для серверов с ролями `storage`, `postgres`, `arangodb` и `search` рекомендуется выделить независимые диски или блочные устройства.

2. Для ролей `postgres`, `arangodb` и `search` монтирование выполняется автоматически во время установки. Путь к смонтированным ролям:

```
/var/lib/docker/volumes/<service_name>
```

Где `<service_name>` — имя роли.

3. Для серверов с ролью `storage` монтирование независимых дисков или блочных дисков автоматически не производится.

3.1 Для хранения данных GlusterFS используется директория: /gluster_bricks/pgs-files. Для обеспечения отказоустойчивости следует использовать монтирование независимых дисков или блочных дисков для этой директории.

Для доступа к хранилищу типа FS со стороны группы сервисов Pythagoras используется точка монтирования /media/storage (с помощью переменной path в файле inventory.). Запрещается изменять точку монтирования /media/storage на /gluster_bricks/pgs-files в связи с уменьшением отказоустойчивости системы и нарушением сихронизации нод сервиса GlusterFS.

3.2 Для хранения данных MinIO используется следующие директории:

– для standalone `/opt/Pythagoras/minio/data/sa0`;

– для кластера `/opt/Pythagoras/minio/data/data[0-n]` (`0-n` — номер используемого диска).

Для обеспечения отказоустойчивости следует использовать монтирование независимых дисков или блочных дисков для этих директорий.

3.3 Рекомендуется использовать форматирование XFS поверх форматирования LVM.

4. При выборе типа хранилища s3 следует ознакомиться с требованиями к конфигурации отказоустойчивости и разделом «Выбор хранилища типа S3».

2.7.4 Общие переменные для CO и PGS

Переменные файлов inventory для CO и PGS, значения которых при установке должны совпадать, приведены в таблицах 11 и 12.

При интеграции CO и PSN переменные указаны в документе «Руководство по настройке».

Начиная с версии 3.3 переменные переименованы, при этом совместимость со старыми переменными сохранена.

Таблица 11 — Общие переменные домена

Имя переменной	Переменная PGS	Переменная CO
Имя домена	DEFAULT_DOMAIN	domain_name
Домен зоны*	ENV	domain_env
** — если домен зоны не используется, то значение переменной указывается ""*		

Таблица 12 — Сводная таблица общих переменных

<table>
<tr><th>Имя переменной</th><th>Расположение в дистрибутиве PGS</th><th>Расположение в дистрибутиве CO</th></tr>
<tr><td>auth_encryption_key</td><td rowspan="11">hosts-sa.yml, hosts-hl.yml, hosts-hl-sa.yaml</td><td rowspan="13">group_vars/co_setup/main.yml</td></tr>
<tr><td>auth_encryption_iv</td></tr>
<tr><td>auth_encryption_salt</td></tr>
<tr><td>fs_token_salt_ext</td></tr>
<tr><td>fs_app_encryption_key</td></tr>
<tr><td>fs_app_encryption_iv</td></tr>
<tr><td>fs_app_encryption_salt</td></tr>
<tr><td>openresty_api_username</td></tr>
<tr><td>openresty_api_password</td></tr>
<tr><td>pgs_rabbitmq_password</td></tr>
<tr><td>fs_app_password</td></tr>
<tr><td>fs_app_login*</td><td>group_vars/all/main.yml</td></tr>
<tr><td>pgs_rabbitmq_user*</td><td>group_vars/all/main.yml</td></tr>
<tr><td colspan="3">* — рекомендуется использовать значение по умолчанию</td></tr>
</table>

При изменении значения переменной `ADMIN_INTERFACE_EXT_PORT` в конфигурации PGS (по умолчанию 443), необходимо добавить следующую переменную в файл `group_vars/co_setup/main.yml`:

```
ADMIN_BASE_URL: "admin-<domain_env>.<domain_name>:<ADMIN_INTERFACE_EXT_PORT>"
```

При изменении значения переменной `API_INTERFACE_EXT_PORT` в конфигурации PGS (по умолчанию 443), следует указать новое значение порта в переменных файла `group_vars/co_setup/main.yml`, используя следующие значения:

```
fs_api_url: "https://pgs-<domain_env>.<domain_name>:\
<API_INTERFACE_EXT_PORT>/pgsapi"
fs_app_url: "https://pgs-<domain_env>.<domain_name>:\
<API_INTERFACE_EXT_PORT>/pgsapi"
fs_card_url: "https://pgs-<domain_env>.<domain_name>:\
<API_INTERFACE_EXT_PORT>/pgsapi"
```

2.7.5 Выбор хранилища типа S3

2.7.5.1 Требования к хранилищу GlusterFS

Хранилище GlusterFS устанавливается в режиме `replicated`, в котором количество нод/серверов в роли `storage` не влияет на потенциально доступное место для PGS. При увеличении количества нод/серверов повышается отказоустойчивость.

Место хранения ограничено размером раздела тома хранения `brick`. При создании раздела `brick` рекомендуется использовать менеджер логических дисков LVM для обеспечения возможности расширения объема дискового пространства.

2.7.5.2 Требования к хранилищу MinIO

Требования к хранилищу MinIO представлены в таблице 13.

Таблица 13 — Конфигурация отказоустойчивости для хранилища S3 MinIO

Конфигурация отказоустойчивости	Количество нод/серверов	Количество независимых дисков*
минимальная	2	2
рекомендуемая	4	4
** — количество независимых дисков на сервер задается с помощью переменной minio_drives_per_node. Если значение для переменной не указанно, при установке значение будет задано автоматически и равно* `2`*.*		

2.7.5.3 Требования к количеству серверов или дисков

При проектировании количества серверов/дисков для s3 следует учитывать требования ПО:

1. В `/opt/Pythagoras/minio/data` должны монтироваться диски с использованием файловой системы XFS.
2. Общее количество дисков включает в себя диски для хранения избыточных данных (MinIO будет использовать половину емкости для избыточного хранения данных) (табл. 14).

Таблица 14 — Распределение дисков для избыточного хранения

Диапазон используемых дисков	Количество дисков для избыточного хранения данных
5 или меньше	2
6 - 7	3
8 или больше	4

3. При использовании более 16 дисков количество будет поделено на меньшие наборы, для которых будут справедливы требования п.2.

4. Требования к распределению дисков п. 3 описаны для настроек MinIO по умолчанию.

Для изменения параметров необходимо воспользоваться инструкцией с сайта производителя:
https://min.io/docs/minio/linux/reference/minio-server/settings/storage-class.html#minio-server-envvar-storage-class

Из-за современной архитектуры серверных CPU существует вероятность деградации производительности при использовании более 8 NVMe на один сервер. При использовании HDD-дисков изменений в работе системы при возрастании количества нод/дисков не замечено.

2.7.6 Настройка дополнительных параметров установки

Дополнительные параметры установки находятся в файле `~/install_pgs/group_vars/all/main.yml`. Менять их без согласования с вендором ПО не рекомендуется.

2.7.7 Оптимизация производительности ArangoDB

В версии 3.1 добавлена настройка управления длительностью запросов для предотвращения повышенного потребления памяти сервисом ArangoDB. Значение установлено по умолчанию, и может быть изменено для более точной настройки работы сервиса в соответствии с таблицей 15. В случае прерывания по таймауту AQL-запроса клиентскому приложению (Aristoteles, Euclid) возвращается код ошибки 410, а запрос сохраняется в журнал событий с кодом ошибки `1500 - ERROR_QUERY_KILLED`.

При работе с переменной рекомендуется минимизировать ее значение во избежание переполнения очереди запросов СУБД и последующего повышенного потребления ресурсов памяти и CPU. При использовании небольших значений могут возникнуть ошибки выполнения длительных миграций и других фоновых AQL запросов, не связанных с работой приложения.

Для серверов в минимальной конфигурации при возникновении ошибок следует одновременно уменьшать значения таймаутов клиента и переменной ARANGODB_MAX_RUNTIME.

Для серверов с большим количеством свободных ресурсов, при возникновении ошибок выполнения длительных запросов допускается увеличить значение переменной.

Таблица 15 — Управление длительностью запросов

Наименование переменной	Расположение переменной	Тип	Единицы измерения	Значение по умолчанию*
ARANGODB_MAX_RUNTIME	~/install_MyOffice_PGS/group_vars/all/main.yaml	Integer	Секунда	61
** — Значение по умолчанию состоит из времени таймаута клиента СУБД для приложений Aristoteles и Euclid + 1 секунда.*				

2.7.8 Настройка резервного копирования

2.7.8.1 Автоматическое резервное копирование

В версии 3.3 добавлен процесс автоматического резервного копирования базы данных ArangoDB, PostgreSQL и файлов docker-compose перед установкой продукта.

При выполнении первичной установки резервное копирование не выполняется. После установки подключается сервис резервного копирования по расписанию.

При обновлении системы будет выполнена проверка работоспособности и автоматическое резервное копирование базы данных.

Для отключения автоматического резервного копирования системы следует запустить установку с ключем `-e backup_enabled=false`. Пример команды:

```
ansible-playbook -i new-cluster-hl.yaml playbooks/main.yml -e
backup_enabled=false
```

Переменная `backup_enabled=false` отключает только автоматическое резервное копирование перед установкой продукта. Инициализация резервного копирования по расписанию добавляется в конце установки PGS.

2.7.8.2 Резервное копирование docker-compose

Независимо от типа установки сохраненные при резервном копировании файлы docker-compose размещаются на первом сервере группы `pythagoras`. При выполнении резервного копирования выполняется поиск по условию pgs-.yaml или pgs-.yml в директории установки (по умолчанию `/opt/Pythagoras`).

Путь хранения файлов на сервере: `<Директория установки>/backup-compose/timestamp запуска>/`

2.7.8.3 Резервное копирование баз данных

При кластерной установке продукта сохраненные при резервном копировании файлы баз данных ArangoDB и PostgreSQL размещаются на первом сервере группы `infrastructure`.

Путь хранения файлов на сервере:

- для базы данных ArangoDB: `<Директория установки>/backups/arangodb/`
- для базы данных PostgreSQL: `<Директория установки>/backups/Postgresdb/`

2.7.8.4 Скрипты резервного копирования

В версии 3.3 добавлена функция автоматического размещения преднастроенных скриптов резервного копирования баз данных. Во время установки продукта скрипты копируются на сервер с ролью infrastrusture в каталог `/opt/Pythagoras/scripts/`

Для настройки и использования скриптов используются переменные, указанные в таблице 16.

Таблица 16 — Настройка скриптов резервного копирования

Наименование переменной	Описание переменной	Тип	Значение по умолчанию
BACKUP_DIR_ARANGODB	Путь для файлов дампов ArangoDB	String	<Директория установки> + '/backups/arangodb/'
BACKUP_DIR_POSTGRES	Путь для файлов дампов Postgres	String	<Директория установки> + '/backups/postgresdb/'
BACKUPS_CRON*	Время выполнения скриптов	String	"0 2 * * *"
backupdb	Отключение/включение опции добавления скриптов	boolean	true
* — *Расшифровка значения по умолчанию: запуск в 02:00 каждый день, месяц, день недели*			

Подготовленные скрипты резервного копирования предназначены для баз данных ArangoDB и Postgres. Рекомендуется выполнить оценку используемых баз данных и самостоятельно подготовить скрипты по резервному копированию для всех данных сервера. Примеры скриптов представлены в документе «Руководство по резервному копированию».

3 УСТАНОВКА

3.1 Порядок запуска установки

Для запуска установки PGS необходимо перейти в каталог установки и выполнить следующую команду:

```
./deploy.sh <hosts.yml> <additional ansible keys>
```

Где:

– `<hosts.yml>` — файл inventory (или путь к нему), сконфигурированный в соответствии с разделом «Настройка параметров установки»;

– `<additional ansible keys>` — дополнительные ключи установки (табл. 17);

При успешном выполнении команды сервисы PGS будут запущены автоматически.

Автоматическое обновление компонентов системы не включено в процесс установки ПО, обновление выполняется вручную администратором.

Таблица 17 — Дополнительные ключи установки

Значение ключа	Описание
-e monitoring_enable=<value>	Устанавливает при значении True или не устанавливает при значении False сервисы мониторинга Prometheus и Grafana. По умолчанию — True
-e content_search_enable=<value>	Устанавливает при значении True или не устанавливает при значении False сервисы для поиска по содержимому документов в облачных редакторах СО. По умолчанию — True
-e kernel_ml_deb_enable=<value>	Включает True или отключает False обновление ядра Debian-based ОС (релизы ветки mainline). По умолчанию — True
-e backup_enabled=<value>	Включает True или отключает False автоматическое резервное копирование баз данных при обновлении. Для первичной установки системы указывать ключ не требуется
-e pgs_update=<value>	Включает True режим обновления, False - установка PGS по умолчанию Для первичной установки системы указывать ключ не требуется

При использовании режима дебага "-vvv" логи установки будут содержать в открытом видео пароли к сервисам и пароль учетной записи администратора, созданной при установке. Файл логов аналогично файлу логов будет содержать подобную информацию. Для сохранения безопасности рекомендуется переместить inventory файл в надежное хранилище, а логи удалить.

3.2 Проверка корректности установки

Для проверки корректности установки необходимо на сервере с ролью `pythagoras` выполнить следующую команду:

```
curl -X POST\
https://pgs-<env>.<default_domain>:<api_interface_ext_port>/pgsapi/?\
cmd=api_version | python3 -m json.tool
```

Где `<env>`, `<default_domain>` и `<api_interface_ext_port>` — переменные, заполненные в соответствии с разделом «Конфигурирование переменных файла hosts.yml».

Пример ожидаемого вывода:

```
{
   "response": {
       "Aristoteles": "19.0.1",
       "success": "true"
    },
    "success": "true"
}
```

Для проверки запуска сервисов PGS необходимо выполнить следующую команду:

```
docker service ls |grep pgs| awk -v OFS='\t' '{print $2, $4}'\
 | column -t
```

Пример вывода:

```
pgs-arangodb_arangodb 1/1
pgs-elasticsearch_elasticsearch 1/1
pgs-etcd_etcd 1/1
pgs-haproxy_haproxy_monitoring 1/1
pgs-keycloak_keycloak 1/1
pgs-nginx_nginx 1/1
pgs-postgres_postgres 1/1
pgs-rabbitmq_rabbitmq 1/1
pgs-redis_redis 1/1
pgs-sisyphus_sisyphussearch 1/1
pgs-sisyphus_sisyphusworker 1/1
pgs_aristoteles 1/1
pgs_dionis 1/1
pgs_epicure 1/1
pgs_euclid 1/1
pgs_flower 1/1
pgs_gateway 1/1
pgs_heraclitus 1/1
pgs_pheidippides 1/1
pgs_polemon 1/1
```

```
pgs_restrictions 1/1
pgs_user_listener 0/0
pgs_users 0/0
```

Сервис `pgs_users` предназначен для интеграции с SSO и настраивается в соответствии с документом «Руководство по настройке интеграции с SSO».

Сервис `pgs_user_listener` предназначен для интеграции внешнего каталога и настраивается в соответствии с документом «Руководство по настройке интеграции с внешними каталогами».

Если интеграции в системе не используются, сервисы `pgs_user_listener` и `pgs_users` после установки не запускаются, при проверке по умолчанию содержат значение `0/0`.

Ошибка запуска сервиса будет представлена значением `0/1`.

4 ОБНОВЛЕНИЕ

4.1 Порядок обновления

В релизе 3.3 добавлена возможность обновления образов docker без полной перестановки продукта.

Функция предназначена для обновления прикладных образов продукта PGS или установки hot-fix версии, где изменяется только образ одного или нескольких приложений.

При обновлении будут выполнены следующие операции:

– Переустановка локального registry;

– Переустановка компонента etcd;

– Обновление docker swarm service pgs_stack и sisyphus;

– Запуск миграции.

Для запуска обновления необходимо добавить дополнительную переменную `pgs_update=true` и `-t update`

```
./deploy.sh <hosts.yml> -e pgs_update=true -t update
```

4.2 Обновление с пропуском версии

При обновлении PGS с пропуском версии (например с 2.5 до 2.8) необходимо выполнить следующую команду:

```
docker exec $(docker ps -q -f name=pgs_aristoteles)\
bash -c "./run_all_migrations.sh"
```

В версии 3.1 для повышения безопасности обновлены настройки хранения и передачи SMTP-паролей для тенантов.

При обновлении версии продукта с 3.0 (или ниже) на версию 3.1 на каждом тенанте необходимо выполнить следующие действия:

1. В административном интерфейсе перейти на вкладку **Организация** > **Основные настройки** > подраздел **Настройки исходящих системных уведомлений**.

2. Ввести значение текущего пароля для SMTP-сервиса и нажать кнопку **Сохранить**.

Без выполнения вышеуказанных пунктов, могут наблюдаться проблемы с отправкой писем и уведомлений пользователям.

5 МИГРАЦИИ

5.1 Настройка связи записей пользователей Keycloak и ArangoDB

Начиная с версии 3.3 для учетной записи пользователя в базе данных ArangoDB добавлено новое поле `external_idm`, которое указывает на его ID в сервисе Keycloak.

Для обновления поля `external_idm` в ArangoDB необходимо использовать следующий скрипт в составе Аристотеля:

```
python3 dev_utils/update_external_idm_for_tenant_users.py <tenant_name>
```

Порядок выполнения скрипта:

1. Идентификация пользователя Keycloak по полю `user_login` в учетной записи пользователя ArangoDB.

2. Обновление поля `external_idm` учетной записи пользователя ArangoDB новым ID пользователя сервиса Keycloak.

6 КАРТА ПОРТОВ

6.1 Карта портов для внутренних соединений

Карта портов для внутренних соединений представлена в таблице 18.

Таблица 18 — Карта портов для внутренних соединений

Группа Ansible	Сервис	Прослушиваемый порт	Порт подключения	Куда обращается (имя swarm service)
nginx	nginx	443/tcp 5673/tcp 15673/tcp 9002/tcp	5672/tcp	rabbitmq
			15672/tcp	rabbitmq
			8854/tcp	polemon
			8000/tcp	gateway
pythagoras	polemon	8854/tcp	-	-
	euclid	8852/tcp	2379/tcp	etcd
			6379/tcp	redis
			8080/tcp	keycloak
			8529/tcp	arangodb
			7002/tcp	dionis
			8080/tcp	restrictions
			5672/tcp	rabbitmq
	aristoteles	8851/tcp	2379/tcp	etcd
			6379/tcp	redis
			8080/tcp	keycloak
			8529/tcp	arangodb
			7000/tcp	sisyphus
			2379/tcp	euclid
			8080/tcp	restrictions
			5672/tcp	rabbitmq
	heraclitus	-	2379/tcp	etcd
			6379/tcp	redis
			8529/tcp	arangodb
			8852/tcp	euclid
			5672/tcp	rabbitmq
			8080/tcp	keycloak
	epicure	-	8529/tcp	arangodb
			6379/tcp	redis
	flower	5555/tcp	8529/tcp	arangodb
			6379/tcp	redis
	dionis	7002/tcp	6379/tcp	redis
			2379/tcp	etcd
	restrictions	8080/tcp	8851/tcp	aristoteles
			5000/tcp	postgres (в режиме кластера)
			5432/tcp	postgres в SA

Группа Ansible	Сервис	Прослушиваемый порт	Порт подключения	Куда обращается (имя swarm service)
			8852/tcp	euclid
	pheidippides	-	6379/tcp	redis
			2379/tcp	etcd
			8851/tcp	aristoteles
	user_listener	-	8852/tcp	euclid
			5000/tcp	postgres в кластере
			5432/tcp	postgres в SA
	user_service	-	8852/tcp	euclid
	pgs_gateway	8000/tcp	8852/tcp	euclid
			8851/tcp	aristoteles
keycloak	keycloak	8080/tcp	8080/tcp	keycloak(в рамках кластера)
			2379/tcp	etcd
			5432/tcp	postgres в SA
			5000/tcp	postgres в кластере
arangodb	arangodb	8529/tcp 8530/tcp	8529/tcp 8530/tcp	arangodb_agent (в режиме кластера)
arangodb_agent	arangodb_agent	8529/tcp 8530/tcp	8529/tcp 8530/tcp	arangodb(в режиме кластера)
search	sisyphus	7000/tcp	2379/tcp	etcd
			9200/tcp	elasticsearch
			5672/tcp	rabbitmq
	elasticsearch	9200/tcp	-	-
rabbitmq	rabbitmq	5672/tcp 15672/tcp	-	-
redis	redis	6379/tcp	-	-
etcd	etcd	2379/tcp	-	-
postgres	postgres	5432/tcp 8008/tcp 5000/tcp 5001/tcp	2379/tcp	etcd
			5432/tcp 8008/tcp 5000/tcp 5001/tcp	postgress
infrastructure	haproxy_postgres	20432/tcp*	5432/tcp	для всех postgres сервисов
	haproxy	20432/tcp*	5432/tcp	для всех postgres сервисов

Группа Ansible	Сервис	Прослушиваемый порт	Порт подключения	Куда обращается (имя swarm service)
		23529/tcp*	8529/tcp	для всех arangodb сервисов
		23080/tcp*	8080/tcp	для всех keycloak сервисов
		30692/tcp*	15692/tcp	для всех RabbitMQ сервисов
		23852/tcp*	8852/tcp	для всех сервисов Euclid
		22851/tcp*	8851/tcp	для всех сервисов Aristoteles
		20555/tcp	5555/tcp	для сервиса flower

* — увеличение количества портов в зависимости от количества серверов в сервисе.

Пример:

Для 3-х сервисов arangodb порты будут: 23529/tcp, 23530/tcp, 23531/tcp.

Для 4-х сервисов arangodb порты будут: 23529/tcp, 23530/tcp, 23531/tcp, 23532/tcp.

** — для работы docker swarm необходимо открыть порты: 2376/tcp, 2377/tcp, 7946/tcp, 7946/udp, 4789/udp.

*** — на момент установки необходимо, чтобы был доступен 5001/tcp с сервера с ролью `infrastructure`.

6.2 Карта внешних портов

Карта портов представлена в таблице 19.

Таблица 19 — Карта внешних портов

Группа Ansible	Сервис	Прослуши-ваемый порт	Порт подключения	Куда обращается (имя swarm service)
Внешние подключения относительно swarm кластера				
nginx	nginx	443/tcp 5673/tcp 15673/tcp 9002/tcp	9000/tcp	Серверы с ролью Storage для коннекта к S3 minio и к другим серверам с s3 сервисом
			514/udp	syslog локально на IP 172.17.0.1

Группа Ansible	Сервис	Прослуши-ваемый порт	Порт подключения	Куда обращается (имя swarm service)
pythagoras	polemon	-	514/udp	syslog локально на IP 172.17.0.1
	eulid	-	9000/tcp	Серверы с ролью Storage для коннекта к S3 minio и к другим серверам с s3 сервисом
			-	PSN(если нужна интеграция)
			514/udp	syslog локально на IP 172.17.0.1
	aristoteles	-	9000/tcp	Серверы с ролью Storage для коннекта к S3 minio и к другим серверам с s3 сервисом
			-	PSN(если нужна интеграция)
			514/udp	syslog локально на IP 172.17.0.1
	heraclitus	-	9000/tcp	Серверы с ролью Storage для коннекта к S3 minio и к другим серверам с s3 сервисом
			514/udp	syslog локально на IP 172.17.0.1
	epicure	-	514/udp	syslog локально на IP 172.17.0.1
	flower	-	514/udp	syslog локально на IP 172.17.0.1
	dionis	-	9000/tcp	Серверы с ролью Storage для коннекта к S3 minio и к другим серверам с s3 сервисом
			514/udp	syslog локально на IP 172.17.0.1
	restrictions	-	514/udp	syslog локально на IP 172.17.0.1
	pheidippides	-	514/udp	syslog локально на IP 172.17.0.1

Группа Ansible	Сервис	Прослуши-ваемый порт	Порт подключения	Куда обращается (имя swarm service)
	user_listener	-	514/udp	syslog локально на IP 172.17.0.1
	user_service	-	514/udp	syslog локально на IP 172.17.0.1
			6379/tcp	Redis на стороне компонента CO
			26379/tcp	Redis Sentinel на стороне компонента CO
	pgs_gateway	-	514/udp	syslog локально на IP 172.17.0.1
keycloak	keycloak	-	514/udp	syslog локально на IP 172.17.0.1
arangodb	arangodb	-	514/udp	syslog локально на IP 172.17.0.1
arangodb_agent	arangodb_agent	-	514/udp	syslog локально на IP 172.17.0.1
search	sisyphus	-	9000/tcp	Серверы с ролью Storage для коннекта к S3 minio и к другим серверам с s3 сервисом
			514/udp	syslog локально на IP 172.17.0.1
	elasticsearch	-	514/udp	syslog локально на IP 172.17.0.1
rabbitmq	rabbitmq	-	514/udp	syslog локально на IP 172.17.0.1
redis	redis	-	514/udp	syslog локально на IP 172.17.0.1
etcd	etcd	-	514/udp	syslog локально на IP 172.17.0.1
postgres	postgress	-	514/udp	syslog локально на IP 172.17.0.1
infrastructure	haproxy_postgres	20432/tcp*	514/udp	syslog локально на IP 172.17.0.1
	haproxy	81/tcp	81/tcp	alertmanager локально на IP 172.17.0.1
		20432/tcp*	514/udp	syslog локально на IP 172.17.0.1
		23529/tcp*	514/udp	syslog локально на IP 172.17.0.1
		23080/tcp*	514/udp	syslog локально на IP 172.17.0.1

Группа Ansible	Сервис	Прослуши-ваемый порт	Порт подключения	Куда обращается (имя swarm service)
		30692/tcp*	514/udp	syslog локально на IP 172.17.0.1
		23852/tcp*	514/udp	syslog локально на IP 172.17.0.1
		22851/tcp*	514/udp	syslog локально на IP 172.17.0.1
		20555/tcp	514/udp	syslog локально на IP 172.17.0.1
За пределами swarm сервиса				
storage	minio	9002/tcp	-	-
infrastructure	alertmanager_ external_port	81/tcp	-	-
	nct_syslog_ng	514/udp	-	-
	grafana	3000/tcp	514/udp	syslog локально на IP 172.17.0.1
			9090/tcp	prometheus (первый сервер с ролью infrastructure)
	prometheus	9090/tcp	514/udp	syslog локально на IP 172.17.0.1
			9100/tcp	pgs_node_export er(все хосты в инвентори)
			9101/tcp	cadvisor (все хосты в группе pgs)
			9187/tcp	pgs_postgres_exp orter(все хосты в группе postgres)
			23080/tcp*	haproxy(обращен ие идет через docker_gwbridge _ip)
			23529/tcp*	haproxy(обращен ие идет через docker_gwbridge _ip)
			30692/tcp*	haproxy(обращен ие идет через docker_gwbridge _ip)
			9121/tcp	redis_exporter(вс е машины в группе redis)

Группа Ansible	Сервис	Прослуши-ваемый порт	Порт подключения	Куда обращается (имя swarm service)
	redis_exporter	9121/tcp	6379/tcp	redis(все машины в группе redis)
	postgresql_ exporter	9187/tcp	20432/tcp*	haproxy(обращение идет через docker_gwbridge _ip)
	cadvisor	9101/tcp	-	-
	node-exporter	9100/tcp	-	-
	Docker-registry***	5001/tcp	-	-

** — увеличение количества портов в зависимости от количества серверов в сервисе.*
Пример:
Для 3-х сервисов arangodb порты будут: 23529/tcp, 23530/tcp, 23531/tcp.
Для 4-х сервисов arangodb порты будут: 23529/tcp, 23530/tcp, 23531/tcp, 23532/tcp.
*** — для работы docker swarm необходимо открыть порты: 2376/tcp, 2377/tcp, 7946/tcp, 7946/udp, 4789/udp. Дополнительно открыт порт 9323/tcp для получения метрик с docker сервиса.*
**** — на момент установки необходимо, чтобы был доступен 5001/tcp с сервера с ролью infrastructure.*

6.3 Рекомендации по открытым портам и доступам

Необходимо обеспечить внешние входящие соединения для серверов с ролью `nginx` (порты перечислены в таблице 19):

– на всех серверах для использования ssh необходимо открыть порт 22/tcp;

– все порты, необходимые для работы ПО Docker Swarm и Docker;

– доступ до сервера с ролью `infrastructure` для просмотра Grafana dashboard;

– доступ с PGS до сервера Redis (CO);

– доступ с PGS до сервера DPS через HAProxy (порт 20003).

Доступ для дополнительного ПО и интеграции:

– при установке GlusterFS необходимо открыть порты: 24007/tcp, 24008/tcp, 49152-49156/tcp для серверов с ролью `storage` и `pythagoras`;

– при интеграции с s3 доступ до s3 хранилища;

– при интеграции с почтовым сервером исходящие подключения к серверу почты.

Для доступа к интерфейсу администратора и API интерфейсу по умолчанию настроен 443 порт.

Для корректной и безопасной работы следует открыть доступ до данных сервисов только со стороны сервера CO, для всех остальных подключений порт к данным сервисам должен быть закрыт.

ПРИЛОЖЕНИЕ А

Известные проблемы и способы их решения

A.1 Бесконечная загрузка во вкладке «Группы» панели администратора

Описание проблемы:

Возникновение бесконечной загрузки во вкладке «Группы» панели администратора PGS.

Для просмотра журнала событий необходимо подключиться к серверу с ролью `infrastructure` и выполнить следующую команду:

```
tail -n 100 /var/log/pgs/<env>.<default_domain>/euclid/critical.log
```

где `<env>`, `<default_domain>` — переменные из файла inventory.

Пример отображения ошибки в журнале событий:

```
RITICAL - 2023-10-31 12:53:04,037 - pgs.euclid - GET /tenants/Default/groups,
Internal server error  500 | - ms
Headers: {'X-FORWARDED-FOR':
...
...
Error: None
Traceback (most recent call last):
  File "/usr/local/lib/python3.11/site-packages/falcon/api.py", line 269, in
__call__
    responder(req, resp, **params)
  File "/usr/local/lib/python3.11/site-
packages/falconswaggerautodoc/schema_decorators.py", line 42, in wrapped
    f(self, *f_args, **f_kwargs)
  File "/opt/Pythagoras/Euclid/endpoints/groups.py", line 91, in on_get
    tenant=req.tenant).data}
                      ^^^^
  File "/opt/Pythagoras/Euclid/serializers/group_serializers.py", line 64, in
data
    return [self._serialize_group(group) for group in self.groups]
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/opt/Pythagoras/Euclid/serializers/group_serializers.py", line 64, in
<listcomp>
    return [self._serialize_group(group) for group in self.groups]
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/opt/Pythagoras/Euclid/serializers/group_serializers.py", line 47, in
_serialize_group
    res["users"] = self.serialize_groupmembers(users)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/opt/Pythagoras/Euclid/serializers/group_serializers.py", line 78, in
serialize_groupmembers
    if "middle_name" in user["attributes"] else ""
                        ~~~~^^^^^^^^^^^^^^^
KeyError: 'attributes'
```

Решение:

1. Запустить на сервере с ролью `keycloack` следующую команду:

```
docker exec $(docker ps -qf name=keycloak)\
 /opt/jboss/keycloak/bin/kcadm.sh create clear-user-cache\
-r <realm> -s realm=<realm> --server\
 http://localhost:8080/auth --realm master\
 --user pgs --password <KEYCLOAK_PASSWORD>
```

2. Запустить на сервере с ролью `pythagoras` следующую команду:

```
docker exec $(docker ps -q -f name=pgs_aristoteles)\
 bash -c "python initializers/RedisInit.py"
```

A.2 Не запускается сервис SisyphusWorker

Описание проблемы:

Не запускается сервис SisyphusWorker и журнал событий содержит следующие ошибки:

```
pgs-sisyphus_sisyphusworker.1.hvjracizck85@    | etcd
      pgs-sisyphus_sisyphusworker.1.hvjracizck85@    | 2379
      pgs-sisyphus_sisyphusworker.1.hvjracizck85@    | 2023/11/13 16:58:24 can't
get arango config with error: client: etcd cluster is unavailable or
misconfigured; error #0: client: endpoint             http://etcd:2379 exceeded
header timeout
      pgs-sisyphus_sisyphusworker.1.r7xptti1p7an@    | etcd
      pgs-sisyphus_sisyphusworker.1.r7xptti1p7an@    | 2379
      pgs-sisyphus_sisyphusworker.1.r7xptti1p7an@    | level=error ts=2023-11-
13T16:58:31Z type=rabbit_wrapper err="dial tcp 10.0.1.14:5672: connect:
connection refused"
      pgs-sisyphus_sisyphusworker.1.r7xptti1p7an@    | panic: Can't create rabbit
wrapper
      pgs-sisyphus_sisyphusworker.1.r7xptti1p7an@    |
      pgs-sisyphus_sisyphusworker.1.r7xptti1p7an@    | goroutine 1 [running]:
      pgs-sisyphus_sisyphusworker.1.r7xptti1p7an@    | main.main()
      pgs-sisyphus_sisyphusworker.1.r7xptti1p7an@
|     /go/src/cmd/worker/main.go:29 +0x80a
      pgs-sisyphus_sisyphusworker.1.kry1r9x8vcdg@    | etcd
      pgs-sisyphus_sisyphusworker.1.kry1r9x8vcdg@    | 2379
```

Необходимо проверить статус работы сервиса RabbitMQ с помощью команды:

```
docker service ps pgsrabbitmq_rabbitmq --format \
'table {{.Name}}\t{{.DesiredState}}'
```

Пример ответа:

```
NAME                          DESIRED STATE
pgs-rabbitmq_rabbitmq.1       Running
```


Проверить, что журнал событий RabbitMQ содержит следующие сообщения:

```
Try to reach pgs-rabbitmq_rabbitmq.3.yq5fexvo3q5g9afw5c8hkotkt
      Try to reach pgs-rabbitmq_rabbitmq.3.yq5fexvo3q5g9afw5c8hkotkt
      Try to reach pgs-rabbitmq_rabbitmq.3.yq5fexvo3q5g9afw5c8hkotkt
      Try to reach pgs-rabbitmq_rabbitmq.3.yq5fexvo3q5g9afw5c8hkotkt
      Try to reach pgs-rabbitmq_rabbitmq.3.yq5fexvo3q5g9afw5c8hkotkt
      Try to reach pgs-rabbitmq_rabbitmq.3.yq5fexvo3q5g9afw5c8hkotkt
```

Решение:

1. Уменьшить количество репликаций для сервиса с помощью команды:

```
docker service scale pgs-rabbitmq_rabbitmq=1
```

2. Выполнить перезапуск последнего docker контейнера сервиса RabbitMQ

```
docker service update pgs-rabbitmq_rabbitmq --force
```

3. Проверить статус работы сервиса SisyphusWorker с помощью команды:

```
docker service ps pgs-sisyphus_sisyphusworker --format \
'table {{.Name}}t{{.DesiredState}}'
```

Пример ответа:

```
NAME                                  DESIRED STATE
pgs-sisyphus_sisyphusworker.1         Running
pgs-sisyphus_sisyphusworker.2         Running
pgs-sisyphus_sisyphusworker.3         Running
```

4. Восстановить кластер для сервиса RabbitMQ с помощью команды:

```
docker service scale pgs-rabbitmq_rabbitmq=3
```

A.3 Ошибка установки сервиса MinIO

Описание проблемы:

Ошибка установки сервиса MinIO возникает при использовании DNS-записей в файле `/etc/hosts` в формате:

```
127.0.0.1 <полное доменное имя хоста>
```

При запуске установки сервиса MinIO контейнеры распределяются по нодам кластера. Обращения между нодами выполняются по полному доменному имени хоста, который в соответствии с записью в файле `/etc/hosts` отправляет запросы на IP-адрес 127.0.0.1. Возникает ошибка скрипта установки при проверке связи между нодами MinIO.

Пример ошибки:

```
Get "http://172.17.0.1:9000/minio/health/live": dial tcp 172.17.0.1:9000:
connect: connection refused
```

Решение:

1. Изменить файл IP-адреса в файле `/etc/hosts` на внешние IP-адреса или удалить DNS-записи из файла.
2. Заменить DNS-записи для серверов с ролью `STORAGE` на IP-адреса.

A.4 Ошибка синхронизации пользователей с AD/OpenLDAP

Описание проблемы:

При попытке синхронизации пользователей с AD/OpenLDAP в журнале событий keycloak появляется следующая ошибка:

```
[LDAPStorageProviderFactory] (executor-thread-17) Sync all users from LDAP to
local store: realm: 9aacad43-4aa0-43e5    -a3c6-dd4a85b6ad0f, federation
provider: pgsldapnew pgs-keycloak_keycloak.1.sb0qpf5mwsde@invsr-pgs-be1    |
2024-07-11 13:24:52,522 ERROR [org.keycloak.services.error.KeycloakErrorHandler]
(executor-thread-17) Uncaught server error: java.lang.NumberFormatException:
Cannot parse null string
```

Решение:

В веб-интерфейсе сервиса Keycloak включить и выключить федерацию pgsldapnew

A.5 Окончание срока действия пароля для пользователя PGS

Описание проблемы:

Авторизоваться вне зависимости от ролевой модели невозможно, в случае если истек пароль для служебного пользователя с именем pgs.

При попытке авторизации возникает сообщение об ошибке **Invalid login or password**.

Срок действия пароля по умолчанию составляет 365 дней.

В журнале событий сервиса Aristoteles возникают следующие ошибки:

```
ERROR - 2024-07-16 16:37:36,306 - pgs.aristoteles - failed to get tenant with
error:
None
Traceback (most recent call last):
  File "/usr/local/lib/python3.11/site-packages/tenant_model/keycloak_realm.py",
line 39, in _get_admin
    admin = KeycloakAdmin(
            ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/site-packages/keycloak/keycloak_admin.py", line
91, in __init__
    ...
    ...
              ^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/opt/Pythagoras/Aristoteles/utils/controllers.py", line 50, in
get_tenant_or_401
    raise NotAuthorizedError()
errors.http_responses.auth_errors.NotAuthorizedErrorERROR - 2024-07-16
16:37:36,310 - pgs.aristoteles - auth,    403 | 115 ms
Headers: {'X-Forwarded-For': '10.5.156.197, 10.115.34.214', 'X-Forwarded-Proto':
'https', 'X-Real-Ip': '10.115.34.214', 'Host': 'a stage host', 'Content-Length':
'52', 'Accept': 'application/json', 'X-Client-Useragent': 'Mozilla/5.0 (X11;
Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0
Safari/537.36', 'X-Forwarded-Host': 'a stage host', 'X-Forwarded-Ssl': 'on',
'Content-Type': 'application/x-www-form-urlencoded', 'User-Agent': 'lua-resty-
http/0.06 (Lua) ngx_lua/10026', 'X-Pgs-Request-Id':
'3e6f41533a1942f0ad1c9fbb4d494997#b517fb1cc3d533ab0d04da7723c3e4c9'}
Parameters: {'login': 'a login of user', 'password': '*****', 'expire': '259000'}
Response headers: {'Content-Type': 'application/json; charset=utf-8', 'x-request-
id': 'zrefyevcwdbwrpi', 'X-pgs-request-id':
'3e6f41533a1942f0ad1c9fbb4d494997#b517fb1cc3d533ab0d04da7723c3e4c9'}
Error: {'error_code': '001', 'error_msg': 'Not Authorised', 'ext_msg': 'Not ad
user', 'success': 'false'}
```

В журнале событий сервиса Keycloak возникают следующие ошибки:

```
Jul 16 16:01:00 a stage stamd keycloak[51172]: 2024-07-16 16:01:00,611 WARN
[org.keycloak.events] (executor-thread-45) type=LOGIN_ERROR, realmId=master, clientId=admin-
cli, userId=null, ipAddress=10.0.0.4, error=resolve_required_actions, auth_method=openid-
connect, grant_type=password, client_auth_method=client-secret, username=pgs
```

Решение:

1. Открыть доступ к сервису Keycloak из внешней сети, выполнив следующую команду:

```
docker service update --publish-add published=8091,target=8080 \
      pgs-keycloak_keycloak
```

2. Открыть веб-интерфейс сервиса Keycloak (адрес по умолчанию http://<DEFAULT_DOMAIN>:8091/auth).

3. Ввести имя пользователя и пароль, указанные при установке. При истекшем пароле будет получено предложение изменить истекший пароль (рис. 1). Пароль пользователя содержится в переменной `KEYCLOAK_PASSWORD`, расположенной в файле inventory (hosts.yaml).

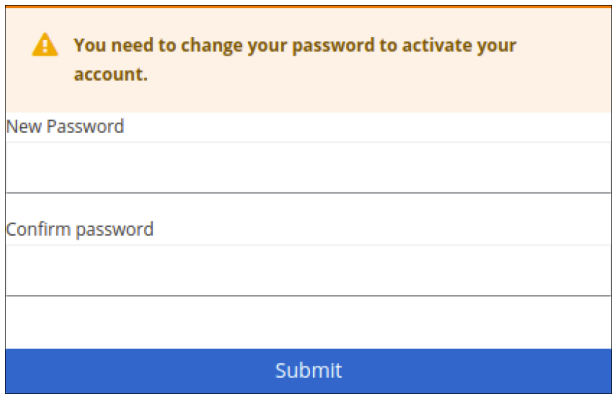

Рисунок 1 — Сообщение о необходимости изменения пароля

4. Увеличение срока жизни пароля (выполняется при необходимости)

4.1 Выбрать Realm Master (рис. 2).

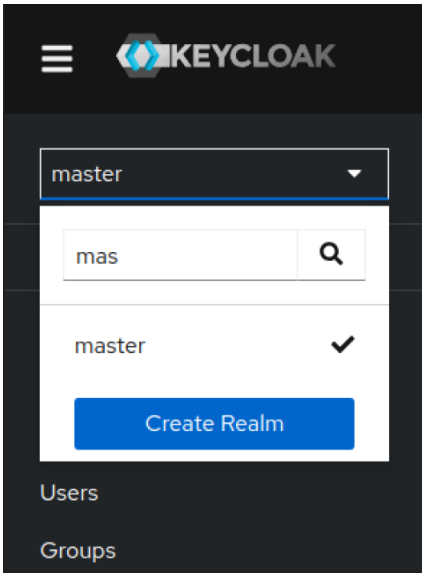

Рисунок 2 — Выбор Realm Master

4.2 Перейти в раздел **Authentication** и открыть вкладку **Policies** (рис. 3).

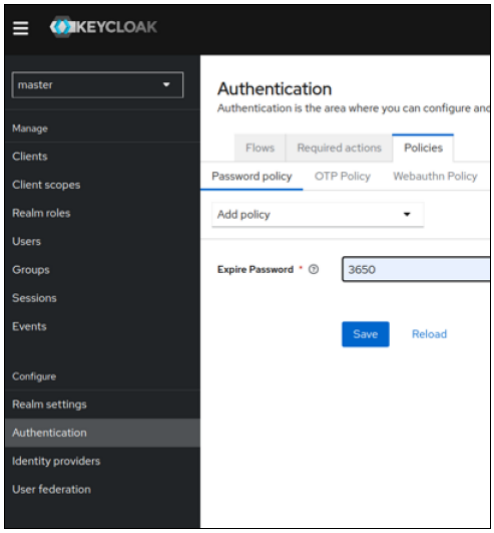

*Рисунок 3 — Вкладка **Policies** в разделе **Authentication***

4.3 Установить срок жизни пароля данного Realm.

A.6 Окончание срока действия пароля для пользователя APP-CO

Описание проблемы:

После авторизации ряд функции недоступен, в случае если истек пароль для служебного пользователя с именем app-co.

Например: невозможно предоставить общий доступ к документу.

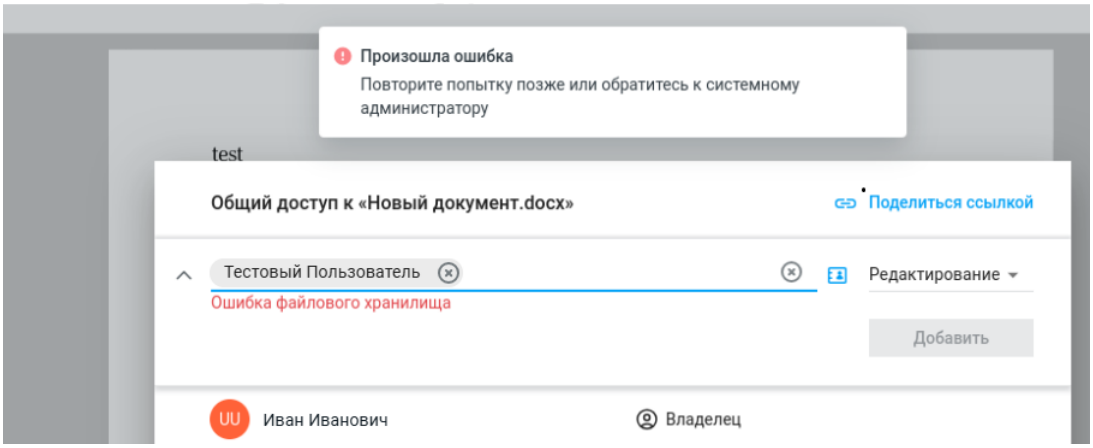

Рисунок 4 — Отсутствие необходимого функционала app-co

В журнале событий сервиса Aristoteles возникают следующие ошибки:

```
ERROR - 2024-07-16 18:35:13,951 - pgs.aristoteles - auth,    400{'login': 'app-
co', 'password': '*****'}{"error_code": "147", "error_msg": "Current password has
expired", "ext_msg": "", "success": "false"}
Traceback (most recent call last):
  File "/usr/local/lib/python3.11/site-packages/tenant_model/model.py", line 915,
in get_token
    token = self.keycloak_realm.oidc.token(username, password)
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/site-packages/keycloak/keycloak_openid.py",
line 207, in token
    return raise_error_from_response(data_raw, KeycloakGetError)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.11/site-packages/keycloak/exceptions.py", line 98,
in raise_error_from_response
    raise error(error_message=message,
keycloak.exceptions.KeycloakGetError: 400:
b'{"error":"invalid_grant","error_description":"Account is not fully set up"}'
...
...
errors.http_responses.auth_errors.ExpiredPasswordErrorERROR - 2024-07-16
18:35:13,955 - pgs.aristoteles - auth,    400 | 299 ms
Headers: {'X-Forwarded-For': '10.5.156.1, 10.115.34.2', 'X-Forwarded-Proto':
'https', 'X-Real-Ip': '10.115.34.2', 'Host': 'a stage stand', 'Content-Length':
'54', 'X-Co-Request-Id': '6c335a927bc24932926098ca1afcb204', 'Content-Type':
'application/x-www-form-urlencoded;charset=UTF-8', 'X-Pgs-Request-Id':
'6c335a927bc24932926098ca1afcb204#7OdtcWoV20oDQvnXefHpyMmz3bEib5GN', 'Accept-
Encoding': 'gzip, x-gzip, deflate, br', 'User-Agent': 'Apache-HttpClient/5.3.1
(Java/21.0.3)'}
Parameters: {'login': 'app-co', 'password': '*****'}
Response headers: {'Content-Type': 'application/json; charset=utf-8', 'x-request-
id': 'xtnkyjtctalwfkg', 'X-pgs-request-id':
'6c335a927bc24932926098ca1afcb204#7OdtcWoV20oDQvnXefHpyMmz3bEib5GN'}
Error: {'error_code': '147', 'error_msg': 'Current password has expired',
'ext_msg': '', 'success': 'false'}
```

Решение:

1. Открыть веб-интерфейс сервиса Keycloak (адрес по умолчанию http://<DEFAULT_DOMAIN>:8091/auth).
2. Ввести имя пользователя и пароль указанные при установке. Пароль пользователя содержится в переменной `fs_admin_password`, расположенной в файле inventory (hosts.yaml).
3. Увеличить срок жизни пароля (выполняется при необходимости).

– выбрать Realm Master (рис. 2);

– перейти в раздел **Authentication** вкладка **Policies** (рис. 3);

– установить срок жизни пароля данного Realm.

ПРИЛОЖЕНИЕ Б

Перечень изменений в документе

В данном приложении представлен перечень изменений относительно даты публикации и версии документа.

28.03.2025	Подготовлен документ версии 1.
17.04.2025	Подготовлен документ версии 2 со следующим изменением: обновлен раздел «Конфигурирование переменных файла hosts.yml» для переменной `nginx_whitelist_network`