МойОфис
Частное Облако 2

В ВАРИАНТЕ ИСПОЛНЕНИЯ ГОСТ

Руководство по установке

СИСТЕМА РЕДАКТИРОВАНИЯ И СОВМЕСТНОЙ РАБОТЫ (СО)

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

«МОЙОФИС ЧАСТНОЕ ОБЛАКО 2» В ВАРИАНТЕ ИСПОЛНЕНИЯ ГОСТ

СИСТЕМА РЕДАКТИРОВАНИЯ И СОВМЕСТНОЙ РАБОТЫ (СО)

РУКОВОДСТВО ПО УСТАНОВКЕ

2.8G

Дата публикации документа:

26.04.2024

На 59 листах

Москва
2024

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

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

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

Таблица 1 — Сокращения и расшифровки

Сокращение, термин	Расшифровка и определение
AD	Microsoft Active Directory, Активный каталог
API	Application Programming Interface, интерфейс программирования приложений
Auth SSO	Single Sign-On, подсистема единого входа (аутентификации и авторизации)
CA	Certificate Authority, удостоверяющий центр для подтверждения подлинности ключей шифрования
CDN	Content Delivery Network, сеть доставки содержимого (в данном случае одна из ролей Auth SSO сервера)
CO	CloudOffice, Облачный Офис, общее название продукта (группы редакторов)
CU	Converter Unit, сервис конвертирования разных форматов файлов
DCS	Document Collaboration Service, сервис редактирования и коллаборации документов на базе кода Core
DNS	Domain Name System, система доменных имен
DU	Document Unit, синоним DCS
EFK	Стек ПО для централизованного сбора и визуализации журналов событий, ElasticSearch + Fluentd + Kibana
ESIA	ЕСИА, Единая Система Идентификации и Аутентификации, информационная система в РФ, обеспечивающая санкционированный доступ для информационного взаимодействия (граждан-заявителей и должностных лиц органов исполнительной власти) к информации, содержащейся в государственных и иных информационных системах
ETCD	Распределенная система хранения конфигурации
FCM	Firebase Cloud Messaging, сервис уведомлений мобильных приложений Google, ранее назывался GCM
FQDN	Fully Qualified Domain Name, полностью определенное имя домена
GCM	Google Cloud Messaging, сервис нотификаций мобильных приложений Google, заменен сервисом FCM
HMS	Huawei Mobile Services, сервис нотификаций мобильных приложений Huawei
Inventory	Файл для настройки Ansible с перечислением ролей и их IP-адресов
IPVS	IP Virtual Server

Сокращение, термин	Расшифровка и определение
JKS	Java Key Store, хранилище ключей и сертификатов, доступных в виртуальном сервере Java
JSON	JavaScript Object Notation, текстовый формат обмена данными, основанный на JavaScript
JVM	Java Virtual Machine, виртуальная машина Java — основная часть исполняющей системы Java, так называемой Java Runtime Environment.
Landing	Стартовая страница
LDAP	Lightweight Directory Access Protocol, облегченный протокол доступа к каталогам
LO	LibreOffice, фильтры которого используются для импортирования устаревших бинарных форматов документов
NPS	Native Process Service, сервис управления нативными процессами (например, конвертацией)
PGS	Pythagoras, сервисы файлового хранилища, работающие по протоколам PGS (Web API, App API, Card API)
Quick launch	Меню быстрого запуска
SMTP	Simple Mail Transfer Protocol, протокол передачи почтовых сообщений
SSH	Secure Shell, «безопасная оболочка»
UI	User Interface, пользовательский интерфейс
URL	Uniform Resource Locator, единый указатель ресурса
UX	User Experience, «опыт пользователя»
ДУ	Директория установки
ООО	Общество с ограниченной ответственностью
ОС	Операционная система
ПК	Персональный компьютер
ПО	Программное обеспечение
УЦ	Удостоверяющий центр

1 ОБЩИЕ СВЕДЕНИЯ

1.1 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 ссерверными 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 иобеспечения 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 намобильных 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».

1.2 Описание архитектуры

Общая архитектурная схема «Системы редактирования и совместной работы (СО)» (далее — СО) приведена на рисунке 1.

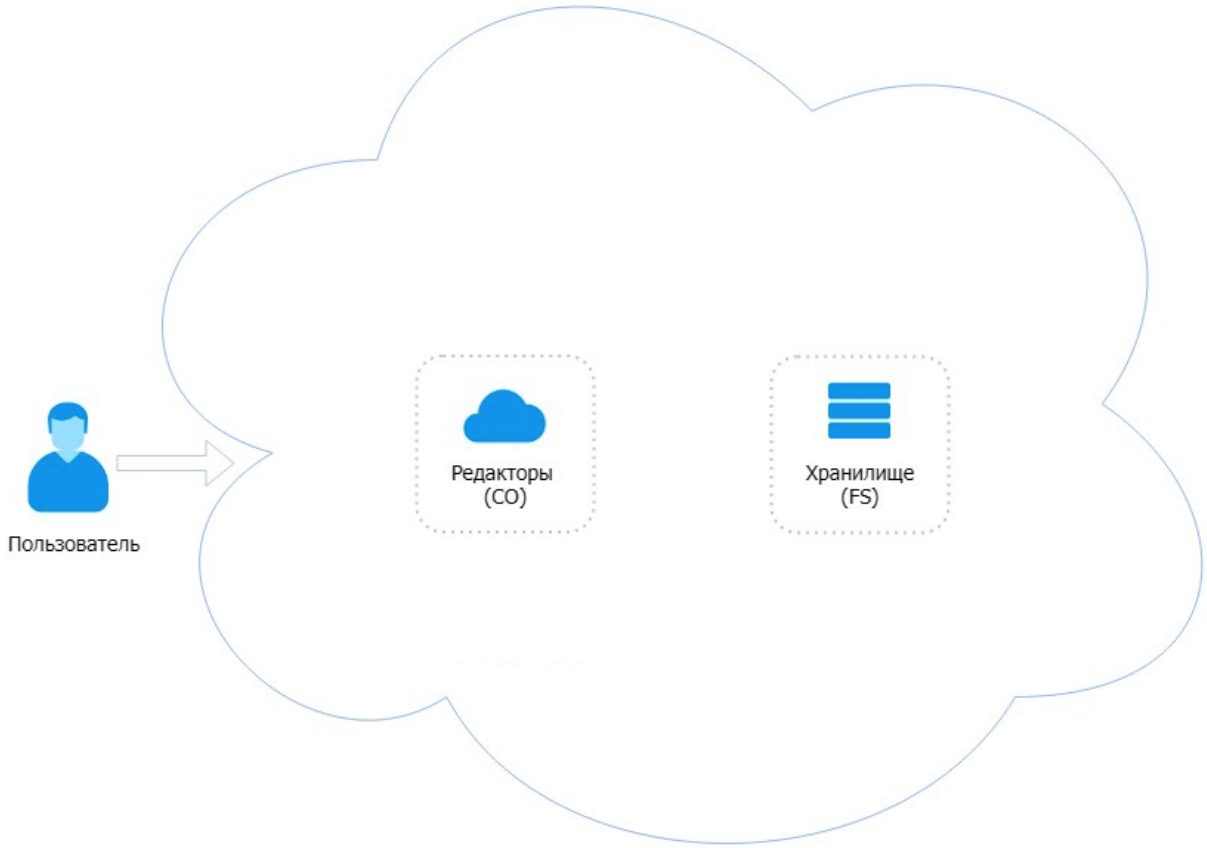

Рисунок 1 — Общая архитектурная схема СО

1.3 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 на уровне эксперта:

– установка Docker;

– запуск/остановка/перезапуск контейнеров;

– работа с реестром контейнеров;

– работа с VMware vSphere ESXi 6.5 и выше;

– получение параметров контейнеров;

– сеть в Docker, взаимодействие приложений в контейнерах;

– решение проблем контейнерной виртуализации.

3. Работа с командной строкой ОС Linux:

– знания в объеме курсов Red Hat RH124, RH134, RH254;

– знания в объеме, достаточном для сдачи сертификационного экзамена Red Hat EX300.

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. Практический опыт администрирования на уровне эксперта:

– Etcd;

– Elasticsearch;

– Prometheus;

– RabbitMQ;

– Redis.

7. Работа с системой автоматизации развертывания Ansible.

1.4 Состав дистрибутива

Комплект поставки ПО предназначен для подготовки инфраструктуры сервера с ролью `operator` и дальнейшей установки СО. Комплект включает в себя:

- исполняемый файл `co_ansible_bin_2.8-gost.run`, предназначенный для установки подсистемы управления конфигурациями;
- исполняемый файл `co_infra_2.8-gost.run`, предназначенный для установки хранилища образов Docker.

1.5 Перечень технической документации

Перечень технической документации, представленный в таблице 1, предназначен для развертывания серверной части, настройки и дальнейшего администрирования продукта «"МойОфис Частное Облако 2" в варианте исполнения ГОСТ».

Комплект документации распространяется на компоненты продукта «"МойОфис Частное Облако 2" в варианте исполнения ГОСТ»:

- Систему редактирования и совместной работы (СО);
- Систему хранения данных (PGS).

Таблица 2 — Перечень технической документации

Наименование документа	Используемые компоненты	Содержание документа
«"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Системные требования»	СО, PGS	Системные и программные требования к продукту
«"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Архитектура»	СО, PGS	Описание архитектуры продукта для выбора типа установки и выделения ресурсов для серверов
«"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Система редактирования и совместной работы (СО). Руководство по установке»	СО	Порядок установки системы редактирования и совместной работы (СО)
«"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Система хранения данных (PGS). Руководство по установке»	PGS	Порядок установки системы хранения данных (PGS)

Наименование документа	Используемые компоненты	Содержание документа
«"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Руководство по настройке»	CO, PGS	Настройка серверов продукта после установки и в ходе эксплуатации системы, а также процессы мониторинга и логирования
«"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Руководство по администрированию»	CO, PGS	Функции управления тенантом в ходе эксплуатации системы Частного облака
«"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Руководство по резервному копированию»	PGS	Порядок резервного копирования баз данных, расположенных в системе хранения данных
«"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Сервисно-ресурсная модель»	CO, PGS	Логическая модель сервиса, описывающая состав и взаимосвязи компонентов (ресурсов), которые совместно обеспечивают предоставление сервиса

1.6 Программные и аппаратные требования

Программные и аппаратные требования к текущей версии ПО указаны в документе «"МойОфис Частное Облако 2". Системные требования».

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 сервера или физические сервера).

1.7.1 Standalone

Конфигурация без отказоустойчивости используется для разработки или демонстрации возможностей продукта (virtual appliance).

Установка в минимальной конфигурации использует три сервера:

– сервер с ролью `operator` для управления процессом установки;

– сервер с ролью `cosa` для установки редакторов и дополнительного ПО;

– сервер с ролью `pgs` для размещения и хранения базовых библиотек и файлов.

1.7.2 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.

1.7.3 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 «"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Системные требования».

2. Если комплект поставляемого ПО включает в себя продукт «МойОфис Почта 2», то необходимо выполнить установку почтового сервера, с помощью сервера с ролью `operator`.

Порядок установки почтового сервера представлен в документе «"МойОфис Почта 2". Руководство по установке почтового сервера»

3. С помощью сервера с ролью `operator` необходимо подготовить инфраструктуру и выполнить установку Системы хранения данных (PGS).

4. С помощью сервера с ролью `operator` необходимо подготовить инфраструктуру и выполнить установку Системы редактирования и совместной работы CO).

5. Выполнить настройку системы для работы с ГОСТ-шифрованием в соответствии с разделом «Настройка ГОСТ сертификатов».

6. Для дальнейшей работы сервер с ролью `operator` не используется, и может потребоваться только для переустановки системы или отдельных сервисов.

7. С помощью документов по настройке, перечисленных в разделе «Перечень технической документации», выполнить необходимые интеграции и установить параметры сервисов.

2 ПОДГОТОВКА К УСТАНОВКЕ

2.1 Подготовка серверов установки

Перед началом установки необходимо ознакомиться с документом «"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Архитектура». В соответствии с типом установки необходимо подготовить нужное количество физических или виртуальных серверов.

На все сервера, предназначенные для развертывания системы, необходимо установить ОС, соответствующую требованиям документа «"МойОфис Частное Облако 2" в варианте исполнения ГОСТ. Системные требования».

2.1.1 Конфигурирование ОС Astra

2.1.1.1 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
```

2.1.1.2 Установка СО 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). Проверить уровень целостности пользователя возможно с помощью команды:

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

2. Установка Ansible и работа СО (версии 2.8G) невозможна при включенном запрете включения бита выполнения. Перед началом установки на всех серверах необходимо выполнить команду:

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

3. Установка Ansible и работа СО (версии 2.8G) невозможна при включенном режиме замкнутой программной среды. Необходимо проверить статус режима с помощью команды:

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


4. При статусе `ACTIVE` перед началом установки на всех серверах необходимо выполнить команду:

```
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. Следует проверить доступность репозиториев, для проверки 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`.

2.1.1.3 Установка СО на максимальном уровне защищенности («Смоленск»)

Для установки СО на максимальном уровне защищенности («Смоленск») необходимо выполнить операции с 1 по 6 раздела «Установка СО на усиленном уровне защищенности («Воронеж»)».

Дополнительно следует отключить обязательную проверку встроенным сканером OpenSCAP образов Docker в файле `~/install_co/group_vars/co_setup/extra_vars.yml`.

Для отключения необходимо добавить к значению переменной `docker_daemon_parameter` следующие параметры:

```
docker_daemon_parameters:
  # TODO: temporarily disable openscap scans until containers are fixed.
  debug: true
  astra-sec-level: 6
```

Запуск установки выполняется с помощью команды:

```
ansible-playbook -i inventory/smolensk.yml playbooks/main.yml \
 -u astra -b
```

2.2 Настройка сетевых соединений

Настройка сетевого соединения должна быть выполнена 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.

Пример настройки сетевого соединения с помощью командной строки в ОС Astra (см. Приложение В).

2.3 Подготовка сервера с ролью operator

2.3.1 Установка дополнительного ПО

В соответствии с документом «"МойОфис Частное Облако 2". 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.

Установка дополнительного ПО может быть выполнена автоматически с помощью скрипта установки. Пример скрипта представлен в Приложении Д.

2.3.2 Установка в сети без выхода в интернет

Для установки СО в локальной сети, без прямого выхода в интернет, необходимо обеспечить доступность дополнительных пакетов ПО. Перечень необходимого ПО приведен в документе «"МойОфис Частное Облако 2". Системные требования».

Для обеспечения доступности необходимо выполнить следующие действия:

– скачать файлы со стандартного репозитория с помощью прямого выхода в интернет;
– разместить файлы на локальном сервере;
– обеспечить доступ к серверу по локальной сети;
– установить ПО и настроить локальный репозиторий (см. Приложение А);
– выполнить замену стандартного репозитория на локальный (см. Приложение Б).

Замена стандартного репозитория на локальный выполняется на сервере с ролью `operator`.

2.3.3 Установка подсистемы управления конфигурациями

Установка выполняется на сервере с ролью `operator`. Порядок действий при установке:

1. Скопировать файл `co_ansible_bin_2.8-gost.run` в корневую директорию пользователя (где `2.8-gost` — имя версии).

2. Запустить скрипт установки:

```
bash co_ansible_bin_2.8-gost.run
```

3. Дать согласие на продолжение установки, нажав на клавишу «Y». Пример запроса:

```
Do you want to continue?      [y/N] y
```

4. После завершения установки на экране пользователя будет отображен список выполненных операций и сообщения. Необходимо убедиться, что список содержит сообщения `[ OK ]` или `[CHANGE]` — это свидетельствует об успешной установке.

При получении сообщения `[FAIL]` необходимо обратиться в техническую поддержку.

2.3.4 Установка хранилища образов Docker

Установка выполняется на сервере с ролью `operator`. Порядок действий при установке:

1. Скопировать файл `co_infra_2.8-gost.run` на сервер с ролью `operator` (где `2.8-gost` – имя версии).

2. Запустить скрипт установки:

```
bash co_infra_2.8-gost.run
```

3. Дождаться проверки целостности файла и его распаковки.

Пример вывода:

```
Verifying archive integrity...      100%        MD5 checksums are OK. All good.
Uncompressing Co Infrastructure Node Package [RELEASE]      100%
```

4. Дать согласие на продолжение установки, нажать на клавишу «Y».

```
Do you want to continue?  [y/N] y
```

5. После завершения работы исполняемого файла на экране пользователя будет отображен список выполненных операций. Необходимо убедиться, что список содержит сообщения `[ OK ]` или `[CHANGE]` — это свидетельствует об успешной установке.

При получении сообщения `[FAIL]` необходимо обратиться в техническую поддержку.

	Для использования других систем контейнеризации необходимо обратиться в техническую поддержку.

2.3.5 Настройка зависимостей Python

На сервере с ролью `operator` зависимости Pyhton указаны в файле `~/install_co/contrib/co/requirements.txt`. Для использования зависимостей из файла необходимо выполнить команду:

```
pip3 install -r ~/install_co/contrib/co/requirements.txt
```

	При автоматической установке модулей Pyhton с помощью скрипта (см. Приложение Д) настройка зависимостей выполняется автоматически

2.4 Создание и размещение сертификатов

2.4.1 Создание ssl-сертификатов

Для обеспечения защищенного соединения между пользователем и сервером СО используется SSL-сертификация. Организации необходимо установить SSL-сертификат на свой сервер, чтобы поддерживать безопасную сессию с браузерами пользователей.

SSL-сертификаты выпускаются доверенным центром сертификации. Браузеры, ОС и мобильные устройства поддерживают список корневых сертификатов доверенных центров сертификации.

В отдельных случаях (например для демонстрации продукта) допускается использование самоподписанного сертификата. Порядок создания самоподписанных сертификатов описан в Приложении Г.

Для упрощения настройки файл переменных `~/install_co/group_vars/co_setup/main.yml` содержит имена сертификатов по умолчанию (секция `TLS cert and key filenames`).

Необходимо использовать сертификаты, выданные центром сертификации для вашей организации, или создать группу новых самоподписанных сертификатов.

2.4.2 Размещение ssl-сертификатов для шифрования

Порядок размещения сертификатов:

1. Разместить сертификат внешнего домена:

```
~/install_co/certificates/server.crt
```

2. Разместить ключ внешнего домена:

```
~/install_co/certificates/server.nopass.key
```

3. Разместить цепочку сертификатов промежуточных центров сертификации (CA) внешнего домена:

```
~/install_co/certificates/ca.pem
```

2.5 Подготовка конфигурационных файлов

Все операции с конфигурационными файлами выполняются на сервере с ролью `operator`.

2.5.1 Порядок установки файлов конфигурации

Директория установки содержит предзаполненные файлы конфигураций, подготовленные для упрощения настройки системы. Директория установки `~/install_co/contrib/co/` содержит два комплекта файлов конфигурации установки: кластерная и standalone.

При обновлении системы допускается использование скопированных и заполненных файлов конфигурации предыдущей версии. Для актуализации значений переменных и параметров установки необходимо ознакомиться со списком изменений в Приложении Д.

В примере показан порядок размещения и настройки файлов конфигурации для кластерной установки:

1. Перейти в каталог `~/install_co/` с помощью команды:

```
cd ~/install_co
```

2. Скопировать файл `~/install_co/contrib/co/ansible.cfg` в корневой раздел директории установки с помощью команды:

```
cp ~/install_co/contrib/co/ansible.cfg ansible.cfg
```

3. Подготовить файл inventory.

Примеры заполненных файлов можно найти в каталоге `~/install_co/contrib/co/`. Скопировать необходимый файл следующей командой:

```
cp ~/install_co/contrib/co/cluster/hosts.yml hosts.yml
```

4. Заполнить файл `hosts.yml` в соответствии с решением об используемой архитектуре.
5. Скопировать ssl-ключи для внешнего домена в каталог `certificates`. Подробнее про размещение ключей можно прочитать в разделе «Размещение ssl-сертификатов для шифрования».
6. Создать в директории групповых переменных `group_vars` каталог для серверов с именем группы установки из файла `hosts.yml` (по умолчанию – `co_setup`).
7. Скопировать в директорию групповых переменных `group_vars` каталог с переменными для заполнения:

```
cp -r ~/install_co/contrib/co/cluster/group_vars/co_setup/* group_vars/co_setup
```

8. Открыть файл `main.yml` из каталога размещения и отредактировать значения параметров в соответствии с разделом «Конфигурирование файла main.yml».
9. Скопировать `run_integration.sh` в корневой раздел директории установки с помощью команды:

```
cp -r ~/install_co/contrib/co/run_integration.sh run_integration.sh
```

2.5.2 Конфигурирование файла hosts.yml

В файл `hosts.yml` вносятся только полностью определенные имена доменов (FQDN). Эти имена будут использоваться во время и после установки системы для обращения к внутренним сервисам.

Директория установки `~/install_co/contrib/co/` содержит два файла inventory, для разных типов установки: кластерной и standalone.

Преднастроенный файл `hosts.yml` содержит примеры заполнения в следующем формате: `co-etcd-1.installation.example.net:`

где:

– `co-etcd-1` — имя сервера для подгруппы `co-etcd`;

– `installation.example.net` — имя домена установки.

Запись в файле `hosts.yml` при использовании группы серверов отличается записью имени сервера: `co-etcd-[1:3].installation.example.net:`

где: `co-etcd-[1:3]` — группа серверов `co-etcd`.

В кластерной конфигурации используется один или нескольких серверов для одной роли.

Пример заполнения файла `hosts.yml`:

```
co:
 children: # Перечень групп
   co_chatbot: #Подгруппа co_chatbot
    hosts:
     co-chatbot-1.installation.example.net: #DNS-имя сервера
    co_etcd:
      hosts:
       co-etcd-[1:3].installation.example.net:
    co_mq:
      hosts:
       co-imc-mq-[1:3].installation.example.net:
```

В конфигурации standalone для всех ролей используется один и тот же сервер.

Пример заполнения файла `hosts.yml`:

```
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:
```

Объединение ролей может применяться в кластерной установке, если ресурсы организации ограничены. Подробнее о выделении ресурсов для установки см. в документе «"МойОфис Частное Облако 2". Архитектура»

В соответствии с выполненными DNS-записями и принятым решением об архитектуре устанавливаемой системы необходимо заполнить файл `hosts.yml`.

2.5.3 Конфигурирование файла main.yml

Для первичной установки системы необходимо скопировать предзаполненный файл конфигурации из директориии `~/install_co/contrib/co/`.

При повторной установке необходимо открыть с помощью текстового редактора файл, расположенный в директории `~/install_co/group_vars/co_setup/main.yml` и изменить значения для обязательных переменных, перечисленных в таблице 5.

Для обеспечения совместной работы CO и PGS необходимо указать одинаковые значения для переменных, перечисленных в разделе «Общие переменные».

Таблица 5 — Основные переменные

Наименование роли	Заполнение обязательно	Описание
Конфигурация Ansible		
ansible_user	-	Имя пользователя, с которым Ansible подключается к хостам по ssh
co_domain_module	-	Строка-шаблон формирования полного доменного имени
co_external_domain	-	Основной домен, на котором будет работать система
domain_env	-	Домен зоны устанавливается в соответствии с разделом «Внешние DNS-записи»
domain_name	+	Имя домена, указывается в соответствии с доменом установки
Конфигурация CA (Центра сертификации)		
ca_main_config.auth_keys.services.key	+	Сгенерировать ключ для доступа к CFSSL API с помощью команды: `"openssl rand -hex 16"`
Конфигурация Docker		
docker_daemon_parameters.insecure-registries	+	Заменить на IPv4 адрес машины оператора и порт 5000

Наименование роли	Заполнение обязательно	Описание
		(например `["10.1.2.3:5000"]`)
docker_image_registry	+	Заменить на IPv4 адрес машины оператора и порт 5000 (например `10.1.2.3:5000`)
bip	-	Адрес сетевого интерфейса (моста) Docker
dns	-	Внутренние DNS-сервера (если не используется unbound)
mtu	-	Размер сетевого пакета сети Docker (может изменяться в виртуальных сетях OpenStack)
Конфигурация ETCD		
etcd_browser_username	-	Имя пользователя для веб-доступа к etcd
etcd_browser_password	+	Пароль пользователя для веб-доступа к etcd
Конфигурация Grafana		
grafana_admin_password	+	Пароль администратора grafana
Конфигурация ELK		
elasticsearch_admin_password	+	Пароль администратора elasticsearch
elasticsearch_admin_password_hash	+	Хеш пароля администратора elasticsearch
elasticsearch_kibana_password_hash	+	Хеш пароля пользователя elasticsearch Kibana
kibana_elasticsearch_password	+	
Конфигурация KEEPALIVED		
keepalived_redis_password	+	Пароль пользователя
keepalived_redis_vip_address	+	IP-адрес подсети серверов кластерной установки
Конфигурация LCS		
lcs_license_key	-	Ключ сервера лицензирования
lcs_server_base_url	-	Ссылка для сервера лицензирования

Наименование роли	Заполнение обязательно	Описание
Конфигурация RabbitMQ		
rabbitmq_federation_enabled	-	Включение федерации RabbitMQ (значение: `true` или `false`)
rabbitmq_users.root.password	+	Пароль для root пользователя RabbitMQ
rabbitmq_users.couser.password	+	Пароль для couser пользователя RabbitMQ
Конфигурация REDIS		
redis_password	+	Пароль для Redis команды AUTH
Конфигурация TLS		
tls_ca_filename	-	Сертификат центра сертификации
tls_cert_filename	-	Сертификат сервера
tls_key_filename	-	Сертификат ключа доступа
Конфигурация Openresty		
openresty_api_username	-	Имя пользователя для доступа к CO Manage API
openresty_api_password	+	Пароль пользователя для доступа к CO Manage API
mail_integration_mode	-	Установка PSN (значения: `none`, `psn2`) по умолчанию: `none`
openresty_mail_oauth2_client_id	-	Идентификатор клиента OAuth2 для интеграции с PSN2
openresty_mail_oauth2_client_secret	-	Секретный ключ клиента OAuth2 для интеграции с PSN2
openresty_mail_oauth2_redirect_uri	-	Адрес перенаправления клиента OAuth2 при интеграции с PSN2
Настройки доступа к PGS		
fs_api_url	+	HTTP ссылка доступа к PGS WebAPI
fs_app_url	+	HTTP ссылка доступа к PGS AppAPI
fs_card_url	-	HTTP ссылка доступа к PGS CardAPI (в предыдущих версиях MailAPI)

Наименование роли	Заполнение обязательно	Описание
fs_app_login	+	Логин пользователя для подключения
fs_app_password	+	Пароль пользователя для подключения
Настройки шифрования (общие для CO и PGS)		
auth_encryption_key	+	Вектор инициализации алгоритма AES-256-CBC, используемого для шифрования mail_session токена
auth_encryption_iv	+	Секретный ключ алгоритма AES-256-CBC, используемого для шифрования mail_session токена
auth_encryption_salt	+	Salt для данных, передаваемых в алгоритм AES-256-CBC, используемый для шифрования mail_session токена
fs_app_encryption_key	+	Вектор инициализации алгоритма AES-256-CBC, используемого для шифрования настроек тенантов, получаемых от PGS
fs_app_encryption_iv	+	Секретный ключ алгоритма AES-256-CBC, используемый для шифрования настроек тенантов, получаемых от PGS
fs_app_encryption_salt	+	Salt для данных, передаваемых в алгоритм AES-256-CBC, используемый для шифрования настроек тенантов, получаемых от PGS
fs_token_salt_ext	+	Salt токена

Наименование роли	Заполнение обязательно	Описание
Настройка PGS		
pgs_rabbitmq_amqp_uri	+	Полное доменное имя сервера PGS для получения уведомлений от PGS
pgs_rabbitmq_password	+	Пароль для RabbitMQ в PGS
pgs_rabbitmq_user	+	Имя пользователя для RabbitMQ в PGS

Если переменная из таблицы 5 отсутствует в примере файла `~/install_co/group_vars/co_setup/main.yml`, допускается добавление переменной вручную

Для генерации паролей и salt рекомендуется использовать утилиту pwgen. Безопасный пароль необходимо генерировать с помощью команды:

```
pwgen <длина пароля> 1
```

где `<длина пароля>` — должна быть не меньше 20 символов.

Для генерации хешей паролей необходимо использовать утилиту htpasswd. Хеш генерируется с помощью команды:

```
htpasswd -bnBC 12 "" <пароль> | tr -d ':\n'
```

Дополнительные переменные перечислены в таблице 6. Для изменения значения необходимо открыть с помощью текстового редактора файл расположенный в директории: `~/install_co/group_vars/co_setup/extra_vars.yml`.

Таблица 6 — Дополнительные переменные

Наименование роли	Заполнение обязательно	Описание
unbound_forward_addresses	-	Список внешних или внутренних DNS, на которые будут отсылаться запросы из unbound

2.5.4 Общие переменные для CO и PGS

Переменные файлов inventory для CO и PGS, значения которых при установке должны совпадать, приведены в таблице 7.

При интеграции CO и PSN переменные указаны в документе «"МойОфис Частное Облако 2". Руководство по настройке».

Таблица 7 — Сводная таблица общих переменных

CO (расположение group_vars/co_setup/main.yml)	PGS (расположение hosts-sa.yml или hosts-hl.yml)
fs_token_salt_ext	FS_TOKEN_SALT_EXT
fs_app_encryption_key	FS_APP_ENCRYPTION_KEY

CO (расположение group_vars/co_setup/main.yml)	PGS (расположение hosts-sa.yml или hosts-hl.yml)
fs_app_encryption_iv	FS_APP_ENCRYPTION_IV
auth_encryption_key	AUTH_ENCRYPTION_KEY
auth_encryption_iv	AUTH_ENCRYPTION_IV
auth_encryption_salt	AUTH_ENCRYPTION_SALT
openresty_api_username	CO_MANAGE_API_USERNAME
openresty_api_password	CO_MANAGE_API_PASSWORD
fs_app_login	APP_ADMIN_LOGIN
fs_app_password	APP_ADMIN_PASSWORD
pgs_rabbitmq_password	RABBITMQ_PASSWORD
fs_app_encryption_salt	FS_APP_ENCRYPTION_SALT
pgs_rabbitmq_user	RABBITMQ_USER

При изменении значения переменной `ADMIN_INTERFACE_EXT_PORT` в конфигурации PGS (по умолчанию 443), необходимо добавить следующую переменную в `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), необходимо добавить новое значение порта в переменные в `co_setup/main.yml`, выполнив следующие команды:

```
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"
```

2.6 Настройка DNS

2.6.1 Внутренние DNS-записи

Внутренние DNS-записи предназначены для установки системы на серверы кластера.

Для всех серверов, перечисленных в файле `hosts.yml` в соответствии с разделом «Конфигурирование файла hosts.yml» необходимо создать DNS-записи. Для создания записей необходимо использовать DNS-сервер вашей организации.

При отсутствии DNS-сервера организации необходимо создать записи на сервере с ролью `operator` в файле `/etc/hosts`.

Пример содержимого файла `/etc/hosts` для установки типа standalone:

```
192.168.1.100 co-infra-1.installation.example.net
```

Пример содержимого файла `/etc/hosts` для кластерной установки:

```
192.168.1.100 co-etcd-1.installation.example.net
192.168.1.101 co-etcd-2.installation.example.net
192.168.1.102 co-etcd-3.installation.example.net
192.168.1.103 co-imc-mq-1.installation.example.net
192.168.1.104 co-imc-mq-2.installation.example.net
192.168.1.105 co-imc-mq-3.installation.example.net
```

Количество записей соответствует количеству используемых физических или виртуальных серверов.

DNS-сервер организации должен содержать аналогичные записи в соответствии с требованиями собственной настройки.

2.6.2 Внешние DNS-записи

Внешние DNS-записи предназначены для подключения пользователей к сервисам.

На DNS-сервере вашей организации необходимо создать записи в соответствии с таблицей 8 или 9.

При отсутствии DNS-сервера организации необходимо создать записи на сервере с ролью `operator` в файле `/etc/hosts` (см. раздел «Внутренние DNS-записи»).

Запрещается использовать в качестве домена зону *.local

Таблица 8 сформирована для параметра `co_domain_module` со значением `{service}.{domain}` (т.е. формирование ссылок через точку к указанному домену).

При формировании записи `{service}.{domain}` переменная `<domain_env>` не используется. В файле `~/install_co/group_vars/co_setup/main.yml` значение переменной должно остаться пустым:

```
domain_env: ""
```

Таблица 8 — Внешние DNS записи со значением `{service}.{domain}`

Имя записи	Тип записи	Значение	Комментарий
auth.<domain_name>	A	IP-адрес, указанный в группе co_lb_core_auth	Количество A записей должно соответствовать количеству серверов
cdn.<domain_name>	CNAME	auth.<domain_name>	
coapi.<domain_name>	CNAME	auth.<domain_name>	
docs.<domain_name>	CNAME	auth.<domain_name>	
files.<domain_name>	CNAME	auth.<domain_name>	
links.<domain_name>	CNAME	auth.<domain_name>	
_https._tcp.<domain_name>	SRV	auth.<domain_name>	Опционально, для мобильных клиентов

Таблица 9 сформирована для параметра `co_domain_module` со значением `{service}-{domain}` (т.е. формирование ссылок через тире к указанному домену).

Таблица 9 — Внешние DNS-записи со значением `{service}-{domain}`

Имя записи	Тип записи	Значение	Комментарии
auth-<domain_env> .<domain_name>	A	IP-адрес, указанный в группе co_lb_core_auth	Количество A записей должно соответствовать количеству серверов
cdn-<domain_env> .<domain_name>	CNAME	auth-<domain_env> .<domain_name>	
coapi-<domain_env> .<domain_name>	CNAME	auth-<domain_env> .<domain_name>	
docs-<domain_env> .<domain_name>	CNAME	auth-<domain_env> .<domain_name>	
files-<domain_env> .<domain_name>	CNAME	auth-<domain_env> .<domain_name>	
links-<domain_env> .<domain_name>	CNAME	auth-<domain_env> .<domain_name>	
_https._tcp-<domain_env> .<domain_name>	SRV	auth-<domain_env> .<domain_name>	Опционально, для мобильных клиентов

2.6.3 Настройка внутренних DNS-записей

Во время установки производится настройка и запуск локального кеширующего DNS-сервера (Unbound) на серверах группы `co_etcd`. Сервер служит для обработки запросов внутри установки и предназначен для работы контейнеров и серверов через соответствующие параметры групповых переменных.

По умолчанию сервера будут перенастроены на работу через Unbound и не будут принимать параметры DNS-серверов по DHCP.

При необходимости Unbound может быть сконфигурирован для работы с внутренними DNS-серверами. По умолчанию Unbound настроен на маршрутизацию запросов на адреса 8.8.8.8 и 8.8.4.4.

DNS-записи, используемые для работы внутри установки, формируются через «.» (точку) относительно вписанного в файл inventory имени сервера. DNS-записи создаются в Unbound автоматически на основе переменной `ansible_default_ipv4`.

Этот параметр можно переопределить двумя способами:

1. Заполнить все адреса вручную на основе примеров в файле групповых переменных, расположенного в следующей директории:

```
~/install_co/group_vars/co_setup/extra_vars.yml.
```

2. Заполнить все необходимые записи на внешнем DNS-сервере без использования Ansible. При подобном варианте необходимо создать «A» — записи для каждого сервера, вписанного в файл `~/install_co/contrib/co/cluster/hosts.yml`, а также CNAME адреса на все поддомены «*» к каждому серверу, вписанному в `hosts.yml`.

Пример заполнения таких записей приведен в таблице 10.

Таблица 10 — Пример заполнения

Имя записи	Тип записи	Значение
co-infra-1	A	10.10.1.110
*.co-infra-1	CNAME	co-infra-1

После настройки Unbound должен быть недоступен из внешней сети.

При использовании `/etc/hosts` для создания DNS-записей необходимо добавить в файл `~/install_co/group_vars/co_setup/extra_vars.yml` все записи, перечисленные в `/etc/hosts`. Например:

```
unbound_local_zones:
      -  type: "transparent"
         zone: "installation.example.net"
         local_data:
              - domain: "co-etcd-1.installation.example.net"
                type: "A"
               ip: "10.1.2.3"
```

2.6.4 Проверка работы DNS на сервере с ролью operator

После настройки необходимо проверить доступность DNS на сервере с ролью `operator`.

При использовании внешнего DNS-сервера необходимо открыть файл `~/install_co/group_vars/co_setup/extra_vars.yml` с помощью текстового редактора и добавить адрес DNS-сервера, изменив IP-адрес:

```
# DNS settings in /etc/resolv.conf
unbound_forward_addresses:
- "127.0.0.1"
- "8.8.8.8"
```

Для проверки соответствия доменного имени IP-адресу сервера необходимо выполнить команду:

```
> dig A co-infra-1.installation.example.net
```


Пример ответа:

```
; <<>> DiG 9.18.1-1ubuntu1.2-Ubuntu <<>> A co-infra-1.installation.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45369
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494;;
QUESTION SECTION:
;co-infra-1.installation.example.net. IN A ;;
ANSWER SECTION:
co-infra-1.installation.example.net. 900 IN CNAME co-infra-
1.installation.example.net.
co-infra-1.installation.example.net. 900 IN A 192.168.0.1
;; Query time: 23 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP) ;; WHEN: Tue Jan 10 15:56:32
MSK 2023
;; MSG SIZE rcvd: 95
```

В ответе необходимо найти секцию `ANSWER SECTION` и проверить, что доменное имя соответствует IP-адресу.

```
co-infra-1.installation.example.net. 900 IN CNAME co-infra-
1.installation.example.net.
co-infra-1.installation.example.net. 900 IN A 192.168.0.1
```

2.6.5 Проверка соединения с PGS

Порядок установки СО (подробнее см. раздел «Порядок установки серверов») предусматривает, что перед развертыванием системы уже выполнена установка системы хранения данных (PGS).

Необходимо проверить доступность сервера `pgs.domain.name` с сервера `operator`.

3 ДОПОЛНИТЕЛЬНЫЕ ПАРАМЕТРЫ УСТАНОВКИ

В разделе представлены дополнительные параметры установки системы. Настройка перечисленных функций не обязательна.

Если специализированные требования к установке отсутствуют, необходимо перейти в раздел «Запуск установки».

3.1 Порядок обновления ядра Linux

При установке ОС на серверы кластера ядро может быть автоматически обновлено до минимальной требуемой версии. По умолчанию ядро обновляется на kernel-lt (LTS) в ОС Redhat-based (CentOS, РЕД ОС). В ОС Debian-based (Ubuntu, Astra) по умолчанию ядро не обновляется. Поддержка других ядер не гарантируется, обратитесь в техническую поддержку за более подробной информацией.

Для отключения обновления в ОС Redhat-based (CentOS, РЕД ОС) при запуске установки необходимо использовать следующую команду:

```
-e kernel_ml_enabled=false
```

Для обновления ядра до kernel-lt (LTS) в ОС Debian-based (Ubuntu, Astra) при запуске установки необходимо использовать следующую команду:

```
-e kernel_ml_deb_enabled=true
```

В ОС Altlinux автоматическое обновление ядра не поддерживается.

3.2 Настройка уведомлений от PGS

Для получения уведомлений о действиях с профилем пользователя на стороне PGS необходимо активировать федерацию одним из перечисленных способов:

1. Со стороны CO необходимо открыть файл `~/install_co/group_vars/co_setup/main.yml` с помощью текстового редактора и выполнить следующие настройки:

- `pgs_rabbitmq_user` — пользователь RabbitMQ PGS;
- `pgs_rabbitmq_password` — пароль пользователя RabbitMQ PGS. Необходимо раскомментировать строку и изменить значение параметра, эквивалентное параметру RABBITMQ_PASSWORD из установки PGS;

2. Настройка возможна с помощью опции запуска скрипта развертывания:

- `rabbitmq_federation_enabled: true` — включение федерации RabbitMQ;

Проверка статуса федерации приведена в документе «"МойОфис Частное Облако 2". Руководство по настройке».

3.3 Настройка префиксов виртуальных хостов

Префиксы виртуальных серверов Nginx (по умолчанию auth, cdn, coapi, docs, files, links) можно изменить с помощью параметров:

– AUTH_PREFIX (префикс адреса приложения авторизации и целевой страницы Auth SSO);

– CDN_PREFIX (префикс адреса CDN);

– COAPI_PREFIX (префикс адреса COAPI);

– DOCS_PREFIX (префикс адреса приложения редакторов);

– FILES_PREFIX (префикс адреса приложения файлового менеджера);

– LINKS_PREFIX (префикс адреса ссылок на документы).

В имени префикса невозможно использовать символы . или _, остальные допустимые символы описаны в RFC 1123.

Записи в DNS должны соответствовать новым префиксам. Настройка префиксов должна производиться совместно с настройками PGS.

3.4 Настройка дополнительных серверов для аудита

Настройка дополнительных Fluentd серверов для сбора событий выполняется с помощью текстового редактора в файле `~/install_co/group_vars/co_setup/main.yml`. Необходимо добавить в файл перечисленные команды, изменив IP-адреса и порты:

```
# LOG servers for the environment
fluentd_server_upstream_log_servers:
- ip: "server_1_ip_address"
port: "24225"
- ip: "server_2_ip_address"
port: "24225"
```

Данная настройка применяется только при использовании в установке роли `log`. Включение функции задается с помощью переменной, указанной в таблице 11.

Таблица 11 — Подключение серверов аудита

Расположение переменной	Наименование переменной	Тип переменной	Значение
group_vars/co_setup/main.yml	common_fluent_logging_enabled	boolean	true / false (по умолчанию)

3.5 Остановка и запуск системы с помощью консольных команд

Для работы с консолью ПО МойОфис администратору системы необходимо обеспечить ssh-доступ к серверам подсистем в контуре установки. Остановка и запуск Системы редактирования и совместной работы (CO), Почты (PSN) и Системы хранения данных (PGS) выполняются отдельно для каждой подсистемы.

Консольные команды для каждого из компонентов:

```
systemctl stop docker
shutdown <option>
```

Первая команда останавливает контейнеризатор, вторая команда позволяет корректно завершить работу сервисов. Ноды сервисов рекомендуется выключать по очереди. Параметр `<option>` позволяет использовать дополнительные параметры выключения, в том числе таймер и опцию перезапуска. Пример (немедленное выключение с остановкой сервисов):

```
shutdown –h now
```

Запуск каждой подсистемы осуществляется при инициализации и запуске аппаратной части программно-технического комплекса.

3.6 Настройка обработки журналов

Настройка обработки журналов (logrotate) в текущей версии ПО не автоматизирована и настраивается самостоятельно администратором.

3.7 Настройка ротации журналов событий в ElasticSearch

Для защиты диска от переполнения записи журнала событий старше 120 дней автоматически удаляются. Процедура использует политики удаления устаревших индексов в ElasticSearch.

Период автоматического удаления (в днях) задается с помощью параметра `es_index_retention_period_days` при развертывании.

3.8 Настройка ГОСТ сертификатов

Поддержка ГОСТ шифрования осуществляется с помощью ПО «КриптоПро CSP». Для полноценной работы необходимо использовать платную серверную лицензию «КриптоПро CSP». При отсутствии лицензии время работы с поддержкой ГОСТ шифрования будет ограничено.

После успешной настройки сертификатов появляется поддержка шифрования каналов связи с клиентскими приложениями в соответствии со стандартом ГОСТ 34.10-2018. При этом сохраняется режим совместимости с клиентами, не поддерживающими ГОСТ шифрование.

Для настройки параметров ГОСТ сертификатов в директории `~/install_co/certificates/gost` необходимо разместить файлы в формате PFX в соответствии с таблицей 12.

Таблица 12 — Файлы для настройки параметров ГОСТ сертификатов

Наименование файла	Описание содержимого	Требования по применению
certkey-gost.pfx	ГОСТ 34.10-2018 сертификат и приватный ключ (без кодовой фразы) на *.<domain_name>	обязательно
certkey-rsa.pfx	RSA сертификат и приватный ключ (без кодовой фразы) на *.<domain_name>	обязательно
roots-gost.pfx	дополнительные доверенные и промежуточные ГОСТ 34.10-2018 сертификаты	при необходимости
roots-rsa.pfx	дополнительные доверенные и промежуточные RSA сертификаты	при необходимости

Для обеспечения безопасности использование самоподписанных сертификатов или сертификатов тестового удостоверяющего центра нежелательно. Допускается применение только при установке в ознакомительных целях.

3.8.1 Конвертирование ГОСТ сертификата

Конвертирование ГОСТ сертификата системы, подключаемой к ESIA, из формата PFX-контейнера КриптоПро в формат BKS, поддерживаемый ESIA-Bridge, осуществляется сторонними утилитами (см. раздел «Настройка работы с ключами ГОСТ Р 34.10-2012» инструкции ESIA-Bridge):

```
java -cp gost-keytool.jar:bcprov-jdk15on-1.62.jar \
ru.reaxoft.gost.Keytool import_pkcs12 \
--srckeystore certkey-esia-gost.pfx \
--srcstorepass "" \
--srckeypass "" \
--srcalias csp_exported \
--destkeystore esia-bridge.bks \
--deststoretype BKS \
--deststorepass "" \
--destkeypass "" \
--destalias gost2012
```

Отличием данной команды от приведенной в официальной инструкции является использование более новой версии библиотеки bcprov, позволяющей работать с хранилищами, созданными новыми версиями OpenSSL и КриптоПро. После конвертации, созданный этой

командой файл `esia-bridge.bks` должен быть размещен в директории `/usr/share/esia-bridge/conf/` взамен существующего.

В конфигурацию ESIA-Bridge должны быть внесены изменения согласно таблице 13.

Таблица 13 — Конфигурация ESIA-Bridge

Наименование переменной	Описание переменной	Примечание
domain	содержит запись в формате "esia-bridge.<domain_name>:9000"	порт 9000 должен быть открыт в настройках сетевого экрана сервера ESIA-Bridge
esia.host	домен среды ESIA, продуктивной или тестовой	
clients	host должен указывать FQDN SSO, то есть auth[-<domain_env>].<domain_name>	лицензия на используемый домен <domain_name>, приобретается отдельно у компании РЕАК СОФТ
id	мнемоника вызывающей системы в ESIA	
name	имя вызывающей системы в ESIA	
cookieDomain	домен СО, на который будет устанавливаться сессионная cookie	, то есть <domain_name>.

После изменения настроек ESIA-Bridge, необходимо перезапустить сервис с помощью команды:

```
systemctl restart esia-bridge
```

Для настройки интеграции с ESIA со стороны СО необходимо открыть файл `~/install_co/group_vars/co_setup/main.yml` с помощью текстового редактора и указать настройки согласно таблице 14.

Таблица 14 — Интеграция с ESIA со стороны СО

Наименование переменной	Значение переменной	Пример адреса
esia_bridge_entrance_uri	точка входа в ESIA-Bridge со стороны СО	http://esiabridge.example.com:9000/blitz/bridge/entrance
esia_bridge_user_uri	адрес, по которому СО запрашивает данные об авторизованном в ESIA пользователе	http://esia-bridge.example.com:9000/blitz/bridge/user

Для настройки интеграции с ESIA со стороны PGS необходимо выполнить настройки тенанта в разделе ESIA. После корректной настройки тенанта в Etcd должны появиться следующие параметры:

```
      "properties": {
            "esia": {
                  "can_esia_auth":1,
                  "support_esia_auth":1
                  },
      ...
      }
```

Запрос к PGS API, который необходимо для этого выполнить, приведен в инструкции по развертыванию PGS.

3.8.2 Настройка ключей шифрования

В целях безопасности в дистрибутиве отсутствуют ключи шифрования по умолчанию.

`fs_app_password`

Пароль PGS пользователя PGSAppAPI. Это же значение указывается в конфигурации установки пакета PGS в переменной inventory `app_admin_password`.

```
fs_app_encryption_key
fs_app_encryption_iv
fs_app_encryption_salt
```

После развертывания данные не сохраняются в etcd, а помещаются в файл `/srv/docker/openresty/properties/auth_config.props` на узлах с ролью `co_lb_core_auth`. Эти значения должны быть идентичны значениям PGS. Для генерации можно использовать следующие команды:

```
openssl enc -aes-256-cbc -k "<password phrase>" -P -md sha256
auth_encryption_key
auth_encryption_iv
auth_encryption_salt
```

Для шифрования токена `mail_session` необходимо задать значения, идентичные значениям PGS. Для генерации необходимо использовать следующую команду:

```
openssl enc -aes-256-cbc -k "<password phrase>" -P -md sha256
fs_token_salt_ext
```

Для авторизации пользователей ESIA в SSO используется значение переменной `fs_token_salt_ext` из файла inventory PGS. Для генерации необходимо использовать произвольную строку символов из набора «Latin1», с учетом регистра символов. Значения должны быть идентичны значениям PGS.

3.8.3 Настройка конфигурационных файлов

При настройке системы для работы с ГОСТ-сертификатами необходимо:

1. Открыть с помощью текстового редактора файл, расположенный в директории `~/install_co/group_vars/co_setup/main.yml` и вручную добавить переменные, представленные в таблице 15

Таблица 15 — Переменные ГОСТ-шифрования

Наименование переменной	Значение	Требования к заполнению
nginx_gost_cacerts_gost_filename	"roots-gost.pfx"	Указывать при использовании доверенных корневых сертификатов ГОСТ В других случаях — строку оставить пустой
nginx_gost_certkey_gost_filename	"certkey-gost.pfx"	Заполнение обязательно
nginx_gost_certkey_rsa_filename	"certkey-rsa.pfx"	Заполнение обязательно
nginx_gost_cacerts_rsa_filename	"roots-rsa.pfx"	Указывать при использовании доверенных корневых RSA В других случаях — строку оставить пустой
nginx_gost_crypto_pro_license	"лицензия крипто про"	
nginx_gost_external_domain	"{{ co_domain_module.format(service='auth', domain=co_external_domain) }}"	

2. Открыть с помощью текстового редактора файл `hosts.yml` и добавить группу `co_nginx_gost`, содержащую тот же набор хостов, что и группа `co_lb_core_auth`.

3.8.4 Обновление ГОСТ-сертификатов

На сервере с ролью `operator` обновить PFX-сертификаты в папке, расположенной `~/install_co/certificates/gost`, после чего выполнить следующую команду:

```
ansible-playbook playbooks/main.yml -t nginx_gost
```

3.9 Карта портов

Карта портов представлена в таблице 16.

Таблица 16 — Карта портов

Группа Ansible	Сервис	Прослушиваемый порт	Порт подключения	Куда обращается
all	node_exporter	9100/tcp		-
	cadvisor	9101/tcp		-
	fluentd_agent	5140/udp, 5160/udp, 5165/udp, 5180/tcp, 24224/tcp, 5185/udp, 2430/tcp		-
	docker	-	5000/tcp	
co	haproxy	20001/tcp, 20002/tcp, 20004-20007/tcp		-
co_lb_core_auth	openresty-lb-core-auth	80/tcp, 443/tcp, 8080/tcp, 8443/tcp, 8888/tcp	20001/tcp, 20002/tcp, 20004/tcp, 20005/tcp, 20007/tcp, 8404/tcp	haproxy
co_chatbot	chatbot	8004/tcp	2002/tcp	haproxy
			24224/tcp	fluentd-agent
co_etcd	etcd	2379/tcp, 2380/tcp		-
	etcd_browser	8001/tcp		-
co_mq	rabbitmq	4369/tcp, 5672/tcp, 15672/tcp, 25672/tcp		-
co_fm	fm	9091/tcp	2379/tcp	etcd
			20004/tcp, 20005/tcp, 20006/tcp, 20007/tcp	haproxy
			24224/tcp, 5180/tcp, 5160/tcp	fluentd-agent
			443/tcp	pgs
co_cvm	cvm	9094/tcp	2379/tcp	etcd
			20005/tcp, 20006/tcp, 20007/tcp	haproxy
			24224/tcp, 5180/tcp, 5160/tcp	fluentd-agent
			443/tcp	pgs

Группа Ansible	Сервис	Прослушиваемый порт	Порт подключения	Куда обращается
co_cu	sdd_cu	9097/tcp	24224/tcp	fluentd-agent
	cu	30000-65535/tcp	26379/tcp	redis_sentinel
			9097/tcp	cu
			24224/tcp	fluentd-agent
co_dcm	dcm	9095/tcp	2379/tcp	etcd
			20001/tcp, 20002/tcp, 20004/tcp, 20005/tcp, 20006/tcp, 20007/tcp	haproxy
			24224/tcp, 5180/tcp, 5160/tcp	fluentd-agent
			443/tcp	pgs
			26379/tcp	redis_sentinel
co_du	du	30000-65535/tcp	26379/tcp	redis_sentinel
			9097/tcp	du
			24224/tcp	fluentd-agent
	sdd_du	9098/tcp	24224/tcp	fluentd-agent
co_jod	jod	9096/tcp	2379/tcp	etcd
			20001/tcp, 20002/tcp, 20004/tcp, 20005/tcp, 20006/tcp, 20007/tcp	haproxy
			24224/tcp, 5180/tcp, 5160/tcp	fluentd-agent
			443/tcp	pgs
			26379/tcp	redis_sentinel
co_nm	nm	9092/tcp	2379/tcp	etcd
			20001/tcp, 20002/tcp, 20004/tcp, 20005/tcp, 20006/tcp, 20007/tcp	haproxy
			24224/tcp, 5180/tcp, 5160/tcp	fluentd-agent
			443/tcp	pgs
			26379/tcp	redis_sentinel
co_pregen	pregen	8002/tcp	24224/tcp	fluentd-agent
	lsyncd	9022/tcp	-	-
co_imc	redis	6379/tcp, 16379/tcp	-	-
	redis_sentinel	26379/tcp	6379/tcp	redis
co_infra	ca	8890/tcp	-	
	nginx	80/tcp, 81/tcp*	9090/tcp	prometheus
			3000/tcp	grafana
			9093/tcp	alertmanager

Группа Ansible	Сервис	Прослушиваемый порт	Порт подключения	Куда обращается
			5601/tcp	kibana
			8001/tcp	etcd_browser
	prometheus	9090/tcp	9093/tcp	alertmanager
			9115/tcp	blackbox_exporter
			9101/tcp	cadvisor
			2379/tcp	etcd
			9100/tcp	node_exporter
			9121/tcp	redis_exporter
	grafana	3000/tcp	-	-
	alertmanager	9093/tcp	-	-
	blackbox_exporter	9115/tcp	-	-
	redis_exporter	9121/tcp	-	-
	elasticsearch	9200/tcp, 9300/tcp	-	-
	kibana	5601/tcp	9200/tcp	elasticsearch
	fluentd_server	5140/udp, 5160/udp, 5165/udp, 5180/tcp, 24224/tcp, 5185/udp, 2430/tcp	-	-
operator	docker-registry	5000/tcp	-	-

***** — только для standalone инсталляции

4 УСТАНОВКА

4.1 Запуск установки

Установка ПО «МойОфис Документы» выполняется с сервера с ролью `operator` с помощью команды:

```
ansible-playbook playbooks/main.yml --diff
```

Скорость установки зависисит от выделенных вычислительных ресурсов. Для обеспечения непрерывности установки рекомендуется использовать дополнительное ПО Screen, Tmux.

В процессе выполнения команды запускаются роли, описанные в разделе «Конфигурирование файла main.yml».

4.2 Проверка корректности установки

Для проверки корректности установки необходимо запустить ПО «МойОфис Документы», выполнив следующие действия:

1. Открыть в поддерживаемом веб-браузере страницу по адресу `auth.[-<domain_env>.]<domain_name>`, настроенному в разделе «Внешние DNS-записи» (например: `auth.installation.example.net`).

2. Войти в систему с помощью учетных данных пользователя или администратора, сконфигурированных во время установки и настройки PGS.

4.3 Запуск интеграционных тестов

После завершения предварительной проверки подсистемы CO, при наличии работающего PGS, необходимо запустить интеграционные тесты.

Запуск тестов необходимо выполнить на сервере с ролью `operator`.

4.3.1 Настройка параметров скрипта запуска

Параметры скрипта запуска интеграционных тестов `~/install_co/run_integration.sh` передаются через переменные окружения. Значения параметров должны соответствовать параметрам установки. Некоторые примеры значений параметров указаны в комментариях скрипта.

Установка параметров интеграционных тестов выполняется с помощью переменных, представленных в таблице 17.

Таблица 17 — Параметры запуска интеграционных тестов

Наименование параметра	Требования к заполнению	Описание	
etcd_browser_url	обязательный	Полный адрес (URL) сервиса Etcd Browser на узле с ролью co_etcd	
etcd_browser_username	обязательный	Имя пользователя etcd-browser	
etcd_browser_password	обязательный	Пароль для etcd-browser	
super_admin	обязательный	Имя пользователя для учетной записи суперадминистратора PGS (для создания тестовых тенантов и их админов)	
super_password	обязательный	Пароль для учетной записи суперадминистратора PGS	
mail_domain	обязательный	Почтовый домен, без @	
pgs_point	обязательный	Полный адрес (URL) PGS сервиса Euclid API	
tag_integration	при необходимости	Тег образа Docker контейнера с интеграционными тестами	
docker_registry	при необходимости	Адрес (FQDN и опционально порт) Docker Registry, где располагается указанный в tag_integration образ	
registry_username	при необходимости	Логин Docker Registry	
registry_password	при необходимости	Пароль Docker Registry	
account_name	при необходимости	Префикс имени домена создаваемого тенанта, не должен содержать . или _	
password	при необходимости	Пароль создаваемого администратора тенанта и всех создаваемых пользователей	
testset	при необходимости	Набор тестов, используются следующие наборы	
		fast	Очистка простых аккаунтов, запуск интеграционных тестов (используется по умолчанию)
		all	Пересоздание простых и корпоративных аккаунтов, запуск интеграционных тестов
ssl_ignore	при необходимости	Игнорировать невалидные или самоподписанные SSL сертификаты	
max_wait	при необходимости	Максимальное время ожидания асинхронных операций в секундах	
log_out	при необходимости	Путь к журналу ошибок тестов	
info_log_out	при необходимости	Логировать в файл вместо stdout	

Наименование параметра	Требования к заполнению	Описание
log_level	при необходимости	Уровень логирования (debug info warn error)
dns	при необходимости	Адрес непубличного DNS-сервера для закрытой установки
tenant_admin	при необходимости	Полный логин администратора тенанта в виде <логин>@<домен-тенанта>[.<окружение>].<домен-установки>. Например, admin@test-deploy.mrt.example.net Поддомен не должен содержать символы «.» и «_». Тенант и пользователи в нем будут созданы перед тестовым запуском. Если передан параметр tenant_admin, переменные mail_domain и account_name игнорируются

4.3.2 Пример запуска интеграционных тестов

```
export ETCD_BROWSER_URL=http://10.0.0.1:8001
export ETCD_BROWSER_USERNAME=user
export ETCD_BROWSER_PASSWORD=pass
export SUPER_ADMIN=pgs
export SUPER_PASSWORD=pgs_pass
export MAIL_DOMAIN=example.com
export PGS_POINT=https://pgs.example.com/adminapi
./run_integration.sh
```

4.4 Диагностика состояния подсистем

4.4.1 Диагностика состояния Nginx

Перечень проверок для диагностики состояния Ngnix указан в таблице 18.

Таблица 18 — Перечень проверок для диагностики Nginx

Тип проверки	Адрес	Примечание
Проверка статуса работы подсистем Auth/SSO и Core	– https://<локальный-адрес-сервера>: 8443/api/manage/core/status – https://<локальный-адрес-сервера>: 8443/api/manage/docs/status	Параметр «all» в ответе должен быть равен строке «OK».
Проверка текущей конфигурации	– https://<локальный-адрес-сервера>: 8443/api/manage/config	
Просмотр журналов доступа и ошибок	– https://<локальный-адрес-сервера>:	В качестве альтернативы используется просмотр

Тип проверки	Адрес	Примечание
системы Auth/SSO (в случае отсутствия сервера с ролью `co_log`)	8443/api/manage/logs/error – https://<локальный-адрес-сервера>: 8443/api/manage/logs/access – https://<локальный-адрес-сервера>: 8443/api/manage/logs/access_full	журналов событий на сервере с ролью `co_lb_core_auth`, по умолчанию место расположения журнала событий: `/srv/docker/openresty/logs/`
Просмотр списка активных сессий и авторизованных пользователей подсистемы Auth/SSO	– https://<локальный-адрес-сервера>: 8443/api/manage/sessions – https://<локальный-адрес-сервера>: 8443/api/manage/users	

Адрес сервера выбирается из указанных в группе `co_lb_core_auth` файла hosts.yml.

Для обеспечения безопасности доступ к порту 8443, ограниченный на стороне Nginx, должен распространяться на локальный сервер и внутренние (частные) сети с адресами по стандарту RFC1918. Категорически запрещается открывать доступ к порту из публичных сетей.

4.4.2 Диагностика состояния Lsyncd

Диагностика состояния Lsyncd применяется только для кластерного режима установки (в standalone конфигурации lsyncd не используется).

Проверить синхронизацию необходимо в журнале событий с помощью команды:

```
docker logs --tail 10 lsyncd
```

Контейнер lsyncd должен быть запущен на всех узлах с ролью `co_lb_core_authre_wopi`. Проверить статус его работы необходимо с помощью команды:

```
cat /srv/docker/lsyncd/conf/lsyncd/lsyncd.status
```

4.4.3 Диагностика состояния RabbitMQ

Проверка статуса очереди сообщений осуществляется через веб-интерфейс RabbitMQ по адресу http://<локальный-адрес-сервера>:15672. Логин и пароль для авторизации используемых в текущей установке.

Адрес сервера выбирается из указанных в группе `co_mq` файла inventory. Предусмотрены возможности проверки состояния кластера RabbitMQ, создания или удаления очереди обмена или отдельных сообщений.

Для проверки федерации RabbitMQ после настройки необходимо использовать веб-интерфейс RabbitMQ CO, расположенный по адресу `http://<локальный-адрес-сервера>:15672/#/federation`

После каждого развертывания и перезагрузки части PGS или CO необходимо проверять, что RabbitMQ (PGS) развернут виртуальный сервер с именем «CO».

При отсутствии виртуального хоста необходимо создать его с помощью панели администратора RabbitUI.

Для доступа к панели администратора используйте доменное имя, указывающее на сервер PGS. Доменное имя формируется на базе зарегистрированного домена установки PGS (см. подробнее документ «"МойОфис Частное облако 2". Система хранения данных МойОфис (PGS). Руководство по установке») и порта «15673»: `http://pgs-<env>.<default_domain>:15673`.

В качестве логина и пароля используются значения переменных `pgs_rabbitmq_user` и `pgs_rabbitmq_password`.

Для обеспечения безопасности доступ к данному порту должен быть ограничен локальным сервером и внутренними (приватными) сетями с адресами по стандарту RFC1918. Категорически запрещается открывать доступ к нему из публичных сетей.

5 ИЗВЕСТНЫЕ ПРОБЛЕМЫ И СПОСОБЫ РЕШЕНИЯ

5.1 Проблема утечек памяти FM при установке standalone

При развертывании СО в режиме standalone необходимо указывать значение переменной `fm_heap_limit` равным 512m. Проверить текущие настройки можно на серверах с ролью `co_fm`:

```
# docker inspect fm | grep Xmx
"JVM_OPTS=-Dmail.mime.encodeparameters=false
-Dmail.mime.encodefilename=true -Dspring.main.banner-mode=off -Xmx512m
-XX:+HeapDumpOnOutOfMemoryError
-XX:ErrorFile=/srv/docker/fm/logs/hs_err.log
-XX:HeapDumpPath=/srv/core_dumps",
```

Для снижения вероятности ошибок при работе FM сервиса во время загрузки, при включенной проверке типов, необходимо увеличить количество памяти для сервиса с помощью команды:

```
fm_heap_limit: 512m
fm_max_memory: 0 (или в два раза больше fm_heap_limit)
```

5.2 Проблема установки модуля python3-libselinux

Описание проблемы

В некоторых случаях в процессе работы установки на ОС Centos, Redos возможно появление следующей ошибки:

```
2023-01-01 12:00:00,001 p=28456 u=root n=ansible | fatal: [
10.100.100.100]: FAILED! => {"changed": false, "msg": "No package
matching 'python3-libselinux' found available, installed or updated",
"rc": 126, "results": ["No package matching 'python3-libselinux' found
available, installed or updated"]}
```

Решение

Выполнить следующую команду и продолжить установку:

```
sed -i 's@python3-libselinux@libselinux-python3@'
./_versions/2.8/collections/ansible_collections/nct/system/roles/python3/vars/R{E
D,edHat}.yml
```

5.3 Решение проблемы с логами

При остановке ротации (архивирования) логов сервисов Nginx или Pregen необходимо обновить политики безопасности на серверах с ролью `openresty-lb-core-auth` и ролью `pregen`.

Обновления политики безопасности выполняются с помощью команды:

```
restorecon -R /srv/docker
```

После обновления политики необходимо проверить ротацию логов через 48 часов.

Например:

```
[root@jenny ~]# cd /srv/docker/openresty/logs/
[root@jenny logs]# ls
access_full.log access_full.log-20231224-1703378461.gz access.log-20231222-
1703205421.gz  error.log error.log-20231224-1703378461.gz
access_full.log-20231221-1703118661.gz  access_full.log-20231225-1703464201
access.log-20231223-1703290201.gz  error.log-20231221-1703118661.gz
error.log-20231225-1703464201  access_full.log-20231222-1703205421.gz
access.log  access.log-20231224-1703378461.gz  error.log-20231222-
1703205421.gz  nginx.pid
access_full.log-20231223-1703290201.gz  access.log-20231221-1703118661.gz
access.log-20231225-1703464201 error.log-20231223-1703290201.gz
```

5.4 Переполнение диска данными мониторинга

Описание проблемы:

Быстрое заполнение диска при установке standalone или для кластерной установки, на узле кластера с ролью `co_infra`.

Решение:

Быстрое заполнение диска может происходить при поступлении большого количества данных мониторинга или логирования, из-за неправильно настроенных политик их хранения.

По умолчанию данные мониторинга располагаются в директории `/srv/docker/prometheus/data`. Время хранения данных задается при установке СО с помощью переменной `prometheus_storage_tsdb_retention_time` (по умолчанию "21d", то есть 21 день).

При переполнении диска данными мониторинга база данных Prometheus может быть повреждена. Для восстановления работоспособности необходимо удалить директорию `/srv/docker/prometheus/data`. После удаления директории следует переустановить роль, ограничив ее опцией `-limit`, только для роли `co_infra` и указав сценарий `playbooks/infra.yml`. Пример команды:

```
ansible-playbook -i playbooks/infra.yml --tags prometheus --limit co_infra
```

Объем данных журнала событий зависит от количества узлов кластера, количества их контейнеров и уровня протоколирования различных сервисов (настраиваются с помощью Etcd). По умолчанию данные журнала событий располагаются в директории `/srv/docker/elasticsearch/data`. Время хранения данных задается при установке СО с помощью переменной `es_index_retention_period_days` (по умолчанию "120", то есть 120 дней).

В случае переполнения диска данными журнала событий, предусмотрено удаление более старые индексов вручную (структуры хранения и поиска данных в объеме 1 дня). Для этого на узле с ролью `co_infra` необходимо выполнить следующие команды:

```
# пароль вводить из переменной elasticsearch_opendistro_admin_password
      curl -k --user admin https://localhost:9200/_cat/indices
      # выбрать индексы, подлежащие удалению, начинающиеся с "co-"
      curl -X DELETE -k --user admin https://localhost:9200/co-<YYYY.MM.DD>
```

Для уменьшения уровня логирования необходимо изменить значения переменных, приведенных в таблице 19.

Таблица 19 — Перечень переменных журнала мониторинга

Наименование переменной	Значение по умолчанию	Значение для уменьшения глубины лога
common_co_log_level	info	warn/error
chatbot_log_level	info	warn/error
cvm_cu_log_level	info	warn/error
cvm_log_level	info	warn/error
dcm_du_log_level	info	warn/error
dcm_log_level	info	warn/error
du_log_level	info	warn/error
du_nps_log_level	info	warn/error
fm_log_level	info	warn/error
sdd_log_level	info	warn/error

Приложение А

Порядок установки и настройки локального репозитория

1. Создать каталог для размещения репозитория с помощью команды:

```
sudo mkdir -p /srv/repo/alse/main
```

2. Примонтировать образ установочного диска (если на компьютере нет каталога /media/cdrom — то создать каталог /media/cdrom) с помощью команды:

```
[ -d /media/cdrom ] || sudo mkdir /media/cdrom
sudo mount /путь_к_ISO-образу /media/cdrom
```

3. Скопировать файлы из образа в каталог репозитория с помощью команды:

```
sudo cp -a /media/cdrom/* /srv/repo/alse/main
```

4. Отмонтировать ISO-образ диска с помощью команды:

```
sudo umount /media/cdrom
```

4.1 Если требуется, выполнить аналогичные действия для базового репозитория (диска со средствами разработки) с помощью команды:

```
sudo mkdir -p /srv/repo/alse/base
[ -d /media/cdrom ] || sudo mkdir /media/cdrom
sudo mount /путь_к_ISO-образу /media/cdrom
sudo cp -a /media/cdrom/* /srv/repo/alse/base
sudo umount /media/cdrom
```

5. Для обновления основного репозитория (основного диска) с помощью команды:

```
sudo mkdir -p /srv/repo/alse/update-main
[ -d /media/cdrom ] || sudo mkdir /media/cdrom
sudo mount /путь_к_ISO-образу /media/cdrom
sudo cp -a /media/cdrom/* /srv/repo/alse/update-main
sudo umount /media/cdrom
```

6. Для обновления базового репозитория (диска с обновлением средств разработки) с помощью команды:

```
sudo mkdir -p /srv/repo/alse/update-base
[ -d /media/cdrom ] || sudo mkdir /media/cdrom
sudo mount /путь_к_ISO-образу /media/cdrom
sudo cp -a /media/cdrom/* /srv/repo/alse/update-base
sudo umount /media/cdrom
```

Приложение Б

Замена стандартного репозитория на локальный

Замена стандартного репозитория на локальный выполняется на сервере с ролью `operator`. Перечисленный порядок действий используется в ОС Astra. Для замены репозитория необходимо:

1. Отключить внешние репозитории, запустив команду:

```
sed -i "s/^/#/" /etc/apt/sources.list
```

2. Добавить локальный внешний репозиторий, запустив команду:

```
tee -a /etc/apt/sources.list << EOF
deb http://$IP_ADDRESS:8081/repository/astra/ 1.7_x86-64 main contrib non-free
deb http://$IP_ADDRESS:8081/repository/astra-ext/ 1.7_x86-64 main contrib non-
free
EOF
```

где `$IP_ADDRESS` — IP-адрес локального сервера для хранения файлов.

3. Обновить индекс репозитория запустив команду:

```
apt update
```

4. Проверить доступность репозитория (произвести поиск произвольного пакета), запустив команду:

```
apt search pwgen
```

5. Убедиться, что в выводе команды присутствует название пакета pwgen. Вывод команды:

```
root@operator:~# apt search pwgen
Sorting... Done
Full Text Search... Done
pwgen/stable 2.08-1 amd64
Automatic Password generation
root@operator:~#
```

6. Настроить менеджер модулей (pip) на использование локального репозитория, запустив команду:

```
tee /etc/pip.conf << EOF
[global]
trusted-host = $IP_ADDRESS
index = http://$IP_ADDRESS:8081/repository/pypi-proxy/pypi
index-url = http://$IP_ADRESS:8081/repository/pypi-proxy/simple
EOF
```

где `$IP_ADDRESS` — IP-адрес локального сервера для хранения файлов.

Приложение В

Настройка сетевых соединений

Пример настройки сетевого соединения с помощью командной строки в ОС Astra.

1. Для проверки необходимо открыть файл с сетевыми настройками с помощью команды:

```
nano /etc/network/interfaces
```

В открывшемся окне редактора проверить наличие следующей строки:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*
```

1.1 Закрыть окно и вернуться к строке терминала.

1.2 Создать новое соединение с помощью команды:

```
sudo nano /etc/network/interfaces.d/eth0
```

Примечание: если на вашем сервере установлены другие редакторы (vim, vi) замените в команде `nano` на другой редактор.

2. В открывшемся окне редактора в зависимости от типа используемого для настроек ввести команду из пункта 2.1 или 2.2.

2.1 При использовании статического IP-адреса необходимо ввести:

```
echo "auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
gateway 192.168.1.1" > /etc/network/interfaces.d/eth0
```

В примере используются произвольные настройки сетевого соединения. Необходимо заменить предложенные настройки (192.168.1.100, 255.255.255.0, 192.168.1.1) на настройки сетевого окружения созданных серверов.

2.2 При использовании DHCP в окне редактора необходимо ввести:

```
echo "auto eth0
iface eth0 inet dhcp" > /etc/network/interfaces.d/eth0
```

Для корректной работы необходимо закрепить IP-адреса за серверами с помощью настроек DHCP-сервера вашего шлюза (коммутатора).

3. После ввода переменных файл сохранить. Повторно открыть файл командой из пункта 1 для проверки.

4. Задать DNS-сервер

```
echo "nameserver 8.8.4.4" > /etc/resolv.conf
```

Адрес DNS-сервера 8.8.4.4 указан произвольно, если в локальной сети существует внутренний DNS-сервер необходимо изменить адрес 8.8.4.4.

5. Применить настройки сетевого соединения

```
sudo systemctl restart networking
```

Повторить выполнение действия для каждого сервера, используемого для установки.

Приложение Г

Порядок создания самоподписанного сертификата

По умолчанию браузеры не доверяют самоподписанным сертификатам, рекомендуется использовать его только для внутренних целей или в целях тестирования.

1. Проверка или установка OpenSSL.

OpenSSL доступен по умолчанию во всех основных дистрибутивах Linux.

Для поиска установленного ПО OpenSSL и проверки версии необходимо выполнить команду:

```
$ openssl version
```

Если вывод с информацией о версии OpenSSL отсутствует — программа не установлена.

Для установки OpenSSL выполните следующую команду:

```
$ sudo dnf install openssl
```

или

```
$ sudo yum install openssl
```

Выбор команды зависит от типа ОС.

2. Создание SSL-сертификата.

Для создания самоподписанного сертификата SSL необходимо использовать следующую команду:

```
$ openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout server.nopass.key -out server.crt
```

С помощью команды будет создан самоподписанный сертификат, который будет действителен в течение 365 дней.

По умолчанию сертификат и файл ключа будут созданы в текущем каталоге (в каталоге, из которого выполняется команда).

Значения флагов команды:

- `req` – сделать запрос на подпись сертификата;
- `-newkey rsa: 4096` – создать ключ RSA длиной 4096 бит. Если не указано иное, по умолчанию будет создан ключ длиной 2048 бит;
- `-keyout` – имя файла закрытого ключа, в котором будет храниться ключ;
- `-out` – указывает имя файла для хранения нового сертификата;
- `-nodes` – пропустить шаг по созданию сертификата с парольной фразой;
- `-x509` – создать сертификат формата X.509;
- `-days` – количество дней, в течение которых сертификат действителен.

Значения полей CSR:

– `C =` – название страны (двухбуквенный код);

– `ST =` – название штата или провинции;

– `L =` – название населенного пункта;

– `O =` – полное название вашей организации;

– `OU =` – название организационной единицы;

– `CN =` – полное доменное имя.

3. Создание закрытого ключа.

Закрытый ключ необходим для подписи вашего SSL-сертификата. Для создания и сохрания закрытого ключа необходимо выполнить команду:

```
$ openssl genrsa -out server.nopass.key
```

Значения флагов команды:

– `genrsa` – создать закрытый ключ RSA;

– `-out` – выходной файл.

По умолчанию закрытый ключ будет храниться в текущем каталоге (в каталоге из которого выполняется команда).

4. Создание запроса на подпись сертификата (CSR).

CSR – информация, отправляемая в удостоверяющий центр. Для создания CSR необходимо выполнить следующую команду:

```
$ openssl req -new -key server.nopass.key -out server.csr
```

где:

– `req` – запрос на подпись сертификата;

– `-new` – новый запрос;

– `-key` – путь, где хранится ваш файл закрытого ключа;

– `-out` – выходной файл.

После запуска команды, показанной ниже, будет создан самоподписаный сертификат, который будет действителен в течение 365 дней.

```
$ openssl x509 -req -days 365 -in server.csr -signkey server.nopass.key -out
server.crt
```

5. Проверка деталей сертификата выполняется с помощью команды:

```
$ openssl x509 -text -noout -in server.crt
```


Приложение Д

Установка дополнительного ПО

Пример скрипта для установки дополнительного ПО на сервере с ролью `operator`.

```
#!/bin/bash

#
# Copyright (c) New Cloud Technologies, Ltd., 2013-2024
#
# You can not use the contents of the file in any way without New Cloud
# Technologies, Ltd. written permission.
# To obtain such a permit, you should contact New Cloud Technologies, Ltd.
# at http://ncloudtech.com/contact.html
#

# Check operator's host and prepare python virtual environment to
# run ansible deploy.

set -e

ANSIBLE_CORE_VERSION=2.11.12
VENV_DIR=~/venv

function die() {
    echo "ERROR: $*"
    exit 1
}

python3 --version 2> /dev/null || die \
"Python 3 was not found, required python version is 3.6+"

if ! python3 -m virtualenv --version; then
    echo \
"Python virtualenv module was not found, trying to install it via pip3..."
    pip3 \
install virtualenv || die \
"Python virtualenv module install failed. Check your python3 installation."
fi

if [[ ! -f "$VENV_DIR/bin/activate" ]]; then
    echo "Preparing virtual env at $VENV_DIR..."
    python3 -m virtualenv "$VENV_DIR" || die \
"Python virtual env creation failed. Check your python3 installation."
fi

source "$VENV_DIR/bin/activate" 2> /dev/null || die \
"Virtual env activation failed. Please remove $VENV_DIR\
and run this script again to recreate it."

echo "Installing required python modules..."
pip install --upgrade pip || die "Python pip module upgrade failed."
pip install --upgrade "ansible-core==$ANSIBLE_CORE_VERSION" || die \
"Python ansible-core module install failed."
pip install --upgrade dnspython jmespath netaddr passlib || die \
"Additional python modules install failed."

echo "Deployment virtual environment was set up successfully."
echo "Please run Ansible to deploy CO:"
echo "source $VENV_DIR/bin/activate; cd $HOME/install_co; \
ansible-playbook playbooks/main.yml"
```

Приложение E

Перечень изменений в текущей версии

Изменения относительно версии 2.6G

1. В файле inventory `hosts.yml` удалены группы `co_service` и `co_lcs`.

2. При кластерной установке отдельные сервера для групп `co_service` и `co_lcs` не выделяются.

Изменения относительно версии 2.7G

В файле inventory `hosts.yml` добавлена группа `co_audit`.

Приложение Ж

Перечень изменений в документе

В данном приложении представлен перечень изменений относительно даты публикации документа.

03.04.2024 Подготовлен документ для версии 2.8G.

26.04.2024 Внесены следующие изменения в документ для версии 2.8G:

– обновлен раздел «Установка дополнительного ПО»;

– обновлен раздел «Настройка зависимостей Python»;

– изменена нумерация Приложения Д на Е;

– добавлено Приложение Д, Ж.