МойОфис®

Частное облако

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

МойОфис Частное облако

Файловое хранилище

РУКОВОДСТВО ПО УСТАНОВКЕ

2019.03

На 117 листах

Москва
2019

МойОфис

Все упомянутые в этом документе названия продуктов, логотипы, торговые марки и товарные знаки принадлежат их владельцам. Товарные знаки «МойОфис» и «MyOffice» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

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

МойОфис

Содержание

МойОфис

Перечень сокращений, терминов и определений

Перечень терминов и определений приведен в таблице 1.

Таблица 1 – Перечень сокращений, терминов и определений

Сокращение, термин	Расшифровка и определение
БД	База данных
ДУ	Директория установки
ЕСИА	Единая система идентификации и аутентификации
ООО	Общество с ограниченной ответственностью
ОС	Операционная система
ПО	Программное обеспечение
СУБД	Система управления базами данных
Хранилище	Файловое хранилище

1 Общие сведения

1.1 Назначение

Файловое хранилище (далее – Хранилище) входит в состав МойОфис Частное облако.

МойОфис Частное облако – продукт для организации виртуальной рабочей среды в государственных организациях и крупных коммерческих предприятиях. Включает редакторы текста, таблиц, презентаций и приложения для управления почтой, календарем и контактами. МойОфис Частное облако позволяет работать в частном облаке на мобильных устройствах и в веб-браузерах.

Подробное описание функций Хранилища приведено в документе «МойОфис Частное облако. Функциональные возможности».

1.2 Требования к квалификации персонала

Администратор Хранилища должен соответствовать следующим требованиям:

- 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);
- опыт работы с подсистемой виртуализации на уровне эксперта:
  - установка **Docker**;
  - запуск / остановка / перезапуск контейнеров;
  - работа с реестром контейнеров;
  - получение конфигурации контейнеров;

- сеть в **Docker**, взаимодействие приложений в контейнерах;
- решение проблем контейнерной виртуализации;

- опыт работы 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;
- опыт работы со службой доменных имен (DNS):
  - знание основных терминов (DNS, IP-адрес и так далее);
  - понимание принципов работы DNS (корневые серверы, TLD-серверы, серверы имен доменов, разрешающий сервер имен и так далее);
  - знание типов записи и запросов DNS;
- знание видов архитектуры, а также основных компонентов инфраструктуры открытых ключей (PKI), к которым относятся:
  - закрытый и открытый ключи;
  - сертификат открытого ключа;
  - регистрационный центр (RA);
  - сертификационный центр (CA);
  - хранилище сертификатов (CR);
- практический опыт администрирования на уровне эксперта:
  - СУБД PostgreSQL;
  - Redis;
  - OpenStack Swift;

- Ansible.

1.3 Системные требования

Перечень системных требований к программному и аппаратному обеспечению приведен в п. 1.3.1 и п. 1.3.2.

1.3.1 Аппаратные требования

Ниже представлены минимальные и рекомендованные требования.

1.3.1.1 Минимальные требования

Минимальные требования для установки Хранилища на оборудовании без поддержки отказоустойчивости и на отказоустойчивом оборудовании приведены в таблицах 2, 3.

Таблица 2 – Минимальные требования (установка без поддержки отказоустойчивости)

Параметр	Значение
Процессор, cpu	8
Оперативная память, Гб	24
Дисковая подсистема, Гб, тип	120, HDD + 40, SSD
Сетевой интерфейс, Мбит/сек	100
Количество серверов	1

Таблица 3 – Минимальные требования (отказоустойчивая установка)

Параметр	fs_fe	fs_be	fs_db	fs_st	fs_infra
Процессор, cpu	1	2	2	2	2

Оперативная память, Гб	2	4	8	4	8
Дисковая подсистема, Гб, тип	20, HDD	20, HDD	40, SSD	40, HDD + 20 SSD	40, HDD
Сетевой интерфейс, Мбит/сек	1	1	1	1	1
Количество серверов	2	2	3	3	1

1.3.1.2 Рекомендованные требования

Рекомендованные требования для установки Хранилища на отказоустойчивом оборудовании приведены в таблице 4.

Таблица 4 – Рекомендованные требования

Параметр	fs_fe	fs_be	fs_db	fs_st	fs_infra
Процессор, cpu	4	8	24	8	8
Оперативная память, Гб	8	16	128	16	32
Дисковая подсистема, Гб, тип	200, HDD	200, HDD	400, SSD	400, SSD + 16000, HDD	1000, SDD
Сетевой интерфейс, Мбит/сек	10	10	10	10	1
Количество серверов	2	4	3	3	1

1.3.2 Программные требования

Требования к программному обеспечению для места оператора и оборудования, на котором производится установка, приведены в таблицах 5, 6.

Таблица 5 – Требования к программному обеспечению для места оператора

Требование	Описание
Поддерживаемые браузеры	• Chrome – не ниже версии 74; • Microsoft Edge – не ниже версии 18; • Mozilla Firefox – не ниже версии 67; • Apple Safari – не ниже версии 12.1; • Яндекс-браузер – не ниже версии 19.6; • Спутник – не ниже версии 4.1; • Opera – не ниже версии 62
Python	v. 2.7+ или v. 3.5+
Модули Python	• jmespath; • jinja2 v.2.10+ (обновление для CentOS можно выполнить с любого репозитория OpenStack: http://mirror.centos.org/centos/7/cloud/x86_64/openstack-queens/ или https://mirror.yandex.ru/centos/7/cloud/x86_64/openstack-queens/); • ansible 2.8.1+

Таблица 6 – Требования к программному обеспечению для оборудования, на котором производится установка

Требование	Описание
ОС	• CentOS 7.5 и выше; • Альт Сервер 8.2; • Astra Linux Common Edition 2.12, релиз "Орел"
Стандартные репозитории ОС	Обязательное подключение всех стандартных репозиториев ОС либо их зеркал во внутренней сети для установок в закрытом контуре
Пакет epel-release	Обязательная установка пакета либо подключение локальной копии репозитория для

	RedHat-based ОС
Репозитории elrepo и docker-ce	Для операционных систем RedHat и CentOS необходимы локальные зеркала репозиториев elrepo (http://elrepo.org) и docker-ce (https://download.docker.com/linux/centos/docker-ce.repo) для установки соответствующих пакетов ядра linux и ПО docker, не входящих в состав поставки
Доступ	Для каждого сервера, на котором выполняется установка, должен быть обеспечен SSH-доступ: • с sudo привилегиями(ALL=(ALL) NOPASSWD: ALL); • без пароля (доступ по ключу)

1.4 Ограничения

1.4.1 Ограничения при выполнении установки на оборудовании без поддержки отказоустойчивости

Режим и последующая настройка параметров на оборудовании без поддержки отказоустойчивости предоставляется в целях демонстрации функциональности Хранилища. Данный режим не поддерживается, не рекомендуется его использовать.

В данном режиме все роли устанавливаются на один физический или виртуальный сервер.

1.4.2 Ограничения при выполнении кластерной установки

Запрещено совмещать серверные роли между собой. Каждый физический или виртуальный сервер должен содержать только одну серверную роль.

1.4.3 Ограничение по работе с файлом inventory

В файле **host.yml** вносятся только доменные имена. Часть логики установщика использует их для формирования доменных имен и адресов сервисов.

1.4.4 Ограничение по работе с подсистемой управления конфигурациями

В подсистеме управления конфигурациями не должно быть конфигурационных файлов самой подсистемы. В том числе конфигурационного файла, который по умолчанию устанавливается с пакетом (например, /etc/ansible/ansible.cfg). Такой файл требуется удалить либо перезаписать образцом из поставляемого ПО. Подробнее см. в https://docs.ansible.com/ansible/latest/reference_appendices/config.html#the-configuration-file

1.4.5 Ограничение по работе с системами виртуализации

Следующие системы виртуализации поддерживаются для обеспечения работы Хранилища:

- VMWare;
- KVM.

1.4.6 Ограничение по работе с DNS

Во время установки Хранилища производится настройка и запуск локального кеширующего DNS-сервера (**unbound**) на машинах роли **fs_db**. Кеширующий DNS-сервер снижает нагрузку на основные DNS-сервера и увеличивает скорость ответа внутри Хранилища на запросы разрешения доменных имен.

	**unbound** используется для запросов внутри Хранилища и подключается только для контейнеров через соответствующие параметры групповых переменных (**group_vars**). **unbound** не должен быть доступен из внешней сети. Это так же означает, что сами сервера будут принимать параметры серверов разрешения имен по настройкам, выставленным при подготовке серверов, и не будут использовать **unbound**. Использование **unbound** необязательно. Если при заполнении файла с параметрами групповых переменных выставляются параметры **Docker** на основные DNS-сервера, **unbound** будет установлен и настроен, но не будет принимать участия в работе Хранилища.

1.5 Рекомендации

1.5.1 Рекомендации по использованию файловых систем

В соответствии с рекомендациями производителей операционных систем рекомендуется:

- для CentOS и RedHat – использовать файловую систему xfs;
- для AltLinux и AstraLinux – использовать файловую систему ext4.

1.5.2 Рекомендации по разбивке дисков

Разбивку дисков рекомендуется выполнять следующим образом:

- для сервера роли **fs_fe** (хранит только статические файлы веб-интерфейса) – не менее 20 Гб для штатной работы ОС;
- для сервера роли **fs_be** (хранит только образы всех запущенных контейнеров) – не менее 20 Гб для штатной работы ОС;
- для сервера роли **fs_db** (хранит базы данных **Redis** и **Postgres** в каталоге /srv/docker) – не менее 10 Гб места на диске для отдельного раздела, выделяемого под каталог /srv/. Рекомендуется использовать SSD;
- для сервера роли **fs_st** (хранятся только файлы подсистемы объектного хранения данных (**Swift**), которая занимает больше всего места в Хранилище. По умолчанию данные этой подсистемы записываются в каталог /srv/node) – не менее 20 Гб места на диске для отдельного раздела, выделяемого под каталог /srv/. Рекомендуется использовать выделенные диски для хранения файлов, их потребуется смонтировать в каталог /srv/node/sd[a-z]. Данный раздел потребуется увеличивать по необходимости с учетом ожидания роста подсистемы объектного хранения данных;
- для сервера роли **fs_infra** (хранятся данные **docker_registry**, сервера мониторинга и оповещения и пакетные репозитории. В будущих релизах будет реализован коллектор журналов и метрик работы системы. Данные

будут храниться в /srv/docker) – не менее 10 Гб места на диске (потребуется создать отдельный раздел под /var/log) для отдельного раздела, выделяемого под каталог /srv/. Потребуется увеличивать раздел по необходимости с учетом ожидания роста нагрузки на Хранилища и соответственного увеличения объема журналов работы Хранилища.

2 Описание архитектуры МойОфис Частное Облако

Общая архитектура МойОфис Частное Облако приведена на Рисунке 1.

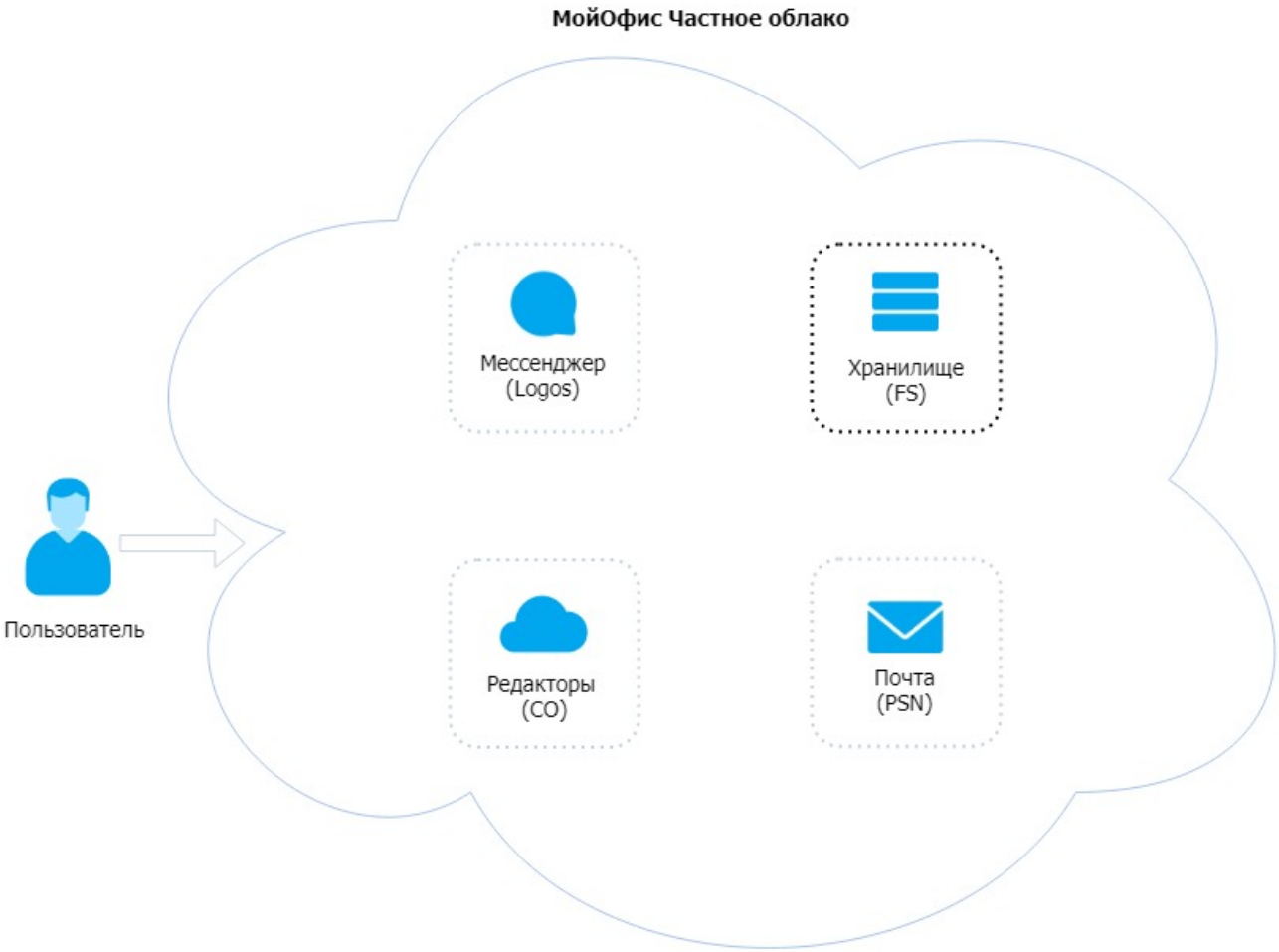

Рисунок 1 – Общая архитектура МойОфис Частное Облако

3 Первичная установка

3.1 Состав дистрибутива

В состав дистрибутива входит программное обеспечение Хранилища.

1. Установщик места оператора (**ansible_bin**).
2. Установщик минимальных параметров роли **fs-infra**, включающий все необходимые образы и пакеты для проведения установки.
3. Файлы tpl.
4. Руководство по установке Файлового хранилища.

3.2 Подготовка к установке

3.2.1 Описание общих ролей подсистемы управления конфигурациями для преднастройки серверов перед установкой

Данные роли описаны в таблице 7.

Таблица 7 – Описание общих ролей **ansible** для преднастройки серверов перед установкой

Наименование роли	Описание
ssh_keys	Добавляет указанные ssh-ключи для выбранных пользователей на сервера группы **play_hosts**
hostname	Устанавливает **hostname** для выбранных серверов группы **play_hosts**[1]
resolv	Производит настройку файла **resolv.conf**
SElinux	Проверяет режим работы **SELinux** и переключает его в режим "enforcing"[2]
yum	Настраивает пакетный менеджер (**yum**), обновляет все пакеты до последней актуальной версии в подключенных репозиториях за исключением **kernel***, **docker-ce***, **container***[2]
package_tools	Добавляет требуемые пакеты для работы Хранилища в целевую ОС

[1] Роль не работает на ОС Astra Linux.
[2] Только для Red-hat based ОС.

locale	Устанавливает параметры **locale** на сервера группы **play_hosts**
timezone	Устанавливает часовой пояс на сервера группы **play_hosts**
sshd	Производит настройку службы **sshd**
chrony	Устанавливает и настраивает службу синхронизации времени **chronyd**
timesyncd	Устанавливает и настраивает службу синхронизации времени[3]
sysctl	Устанавливает требуемые параметры ядра на сервера группы **play_hosts**
limits	Настраивает параметры лимитов на сервера группы **play_hosts**
kernel_ml	Устанавливает пакет **kernel_ml** последнего доступного ядра[3]
sensu	Устанавливает и настраивает клиентскую часть **sensu**, подключает стандартную подписку для мониторинга ресурсов сервера
rsyslog	Устанавливает и настраивает сервис сбора журналов
docker_service	Устанавливает и настраивает **Docker**, подключает к **docker_registry**
iptables	Устанавливает и настраивает службы межсетевого экрана с параметрами, требуемыми для конкретной роли

[3] Роль не работает на ОС Astra Linux.

Роли, используемые для подготовки Хранилища, описаны в таблице 8.

Таблица 8 – Описание ролей, используемых при подготовке Хранилища

Наименование роли	**Описание**
check_installation	Проверяет, что параметры, установленные на этапе подготовки серверов, были установлены правильно
keepalived на fs_db группу	Устанавливает и запускает службу, реализующую протокол VRRP[4]
unbound	Устанавливает и настраивает кеширующий DNS-сервер
ETCD	Устанавливает отказоустойчивую базу данных **ETCD**
stolon на fs_db группу	Устанавливает и настраивает СУБД PostgreSQL с подсистемой отказоустойчивости **stolon**
redis на fs_db группу	Устанавливает и настраивает прокси **Redis**
swift	Устанавливает и настраивает подсистему объектного хранения данных
stolon на fs_be группу	Устанавливает и настраивает прокси до СУБД PostgreSQL с подсистемой отказоустойчивости **stolon**
redis на fs_be группу	Устанавливает и настраивает прокси **Redis** с подсистемой отказоустойчивости **sentinel**
postfix	Устанавливает роль для развертывания почтового сервера (**MTA**)
fleet	Устанавливает и настраивает менеджер сервисов с распределенной системой инициализации
fs (tag fs_setup)	Устанавливает и настраивает все необходимые сервисы Хранилища
keepalived на fs_fe группу	Устанавливает и настраивает запускает службу, реализующую протокол VRRP[4]
nginx	Устанавливает и настраивает веб-сервер (**nginx**)
confd	Устанавливает шаблоны и конфигурации для

[4] Параметр используется для отказоустойчивой установки.

	настроенных сервисов
fs (tag fs_accounts)	Создает необходимые учетные записи для приложений (co, psn, logos)

3.2.2 Подготовка инфраструктуры установки

Для подготовки инфраструктуры установки должны быть проведены следующие действия:

- установка хранилища образов **Docker** (**docker_registry**) и пакетного репозитория;
- установка подсистемы управления конфигурациями (**Ansible**).

Подробная информация о выполнении данных действий приведена в пп. 3.2.2.1 и пп. 3.2.2.2.

3.2.2.1 Установка хранилища образов Docker (docker_registry) и пакетного репозитория

Установка производится на сервере с ролью **fs_infra**. Перед началом установки проверить, что вход выполнен под пользователем **root**.

Этапы установки:

1. Скопировать файл fs_infra_*.run на сервер.
2. Запустить скрипт установки: bash fs_infra_[RELEASE].run, где RELEASE – имя релиза.
3. Согласиться на продолжение установки, нажать на клавишу "Y".

Администратору отобразится:

```
[root@fs-infra-01 ~]# ./fs_infra_releaseXX.run
Verifying archive integrity...  100%   All good.
Uncompressing FS Infrastructure Node Package release38  100%
Welcome to MyOffice Infrastructure Installer
This script is meant to be used on Infrastructure Server (db1 by default)

Do you want to continue? [y/N] y
Check if script is running as superuser                                 [  OK  ]
```

```
Check if Operating System is compatible                          [ OK ]
Ensure that yum-utils is installed                               [ OK ]
Ensure that device-mapper-persistent-data is installed           [CHANGE]
Ensure that lvm2 is installed                                    [CHANGE]
Ensure that docker package is availiable                         [CHANGE]
Ensure that docker is installed                                  [CHANGE]
Ensure that docker is started                                    [CHANGE]
Ensure that docker is enabled                                    [CHANGE]
Test if docker is availiable                                     [ OK ]
Ensure that python-setuptools is installed                       [ OK ]
Ensure that registrytool is installed                            [CHANGE]
Ensure that core images are imported                             [CHANGE]
Ensure that registry configuration directory is present          [CHANGE]
Ensure that registry configuration file is present               [CHANGE]
Ensure that registry systemd unit is present                     [CHANGE]
Ensure that docker-registry.service is loaded to systemd         [ OK ]
Ensure that docker-registry is started                           [CHANGE]
Ensure that docker-registry is enabled                           [ OK ]
Ensure that images are present in registry                       [CHANGE]
Ensure that fs-yumrepo systemd unit is present                   [CHANGE]
Ensure that fs-yumrepo.service is loaded to systemd              [ OK ]
Ensure that fs-yumrepo is started                                [CHANGE]
Ensure that fs-yumrepo is enabled                                [ OK ]
```

После этого установка хранилища образов **Docker** (**docker_registry**) и пакетного репозитория будет завершена.

3.2.2.2 Установка подсистемы управления конфигурациями (Ansible)

Установка производится на рабочем месте оператора. Проверить, что:

- вход выполнен под пользователем **root** или под пользователем **sudo** с привилегиями **yum (dnf)**;
- машина, на которой выполняется установка, соответствует требованиям, указанным в документе "МойОфис Частное облако. Системные требования";
- с выбранного сервера есть возможность доступа по SSH до других серверов, на которых выполняется установка;
- подсистема управления конфигурациями установлена, другие конфигурационные файлы **Ansible** не присутствуют в Хранилище.

Этапы установки:

1. Скопировать файл ansible_bin_releaseXX.run в домашнюю директорию пользователя **root.**
2. Запустить файл ansible_bin_releaseXX.run.
3. Согласиться на продолжение установки, нажать на клавишу "Y".

Администратору отобразится:

```
[root@fs-ansible-installer ~]# ./ansible_bin_XX.run
Verifying archive integrity... 100% All good.
Uncompressing FS Ansible Package releaseXX 100%

Welcome to MyOffice Ansible Installer version XX
This script is meant to be used on operator workstation
Do you want to contiune? [y/N] y

Ensure that ucs-storm is installed                                    [CHANGE]
Ensure that version directory is present                              [CHANGE]
Ensure that version release14 is present                              [CHANGE]
Set release14 as latest                                                [ OK ]
Create roles symlink                                                   [ OK ]
Create playbooks symlink                                               [ OK ]
Create group_vars directory                                            [ OK ]
Create group_vars/all symlink                                          [ OK ]
Create group_vars/fs symlink                                           [ OK ]
Create host_vars directory                                             [ OK ]
```

После этого установка подсистемы управления конфигурациями будет завершена.

3.2.2.3 Установка Хранилища с машины оператора

Этапы установки:

1. Перейти в каталог ~/install_fs/[5] с помощью команды:

```
[root@fs-installer ~]# cd ~/install_fs
```

[5] Далее в отношении ДУ используются относительные пути.

2. Скопировать файл contrib/fs/ansible.cfg в корневой раздел <ДУ> с помощью команды:

```
[root@fs-installer ~]# cp contrib/fs/ansible.cfg .
[root@fs-installer ~]#
```

3. Подготовить файл **inventory**. Примеры заполненных файлов можно найти в каталоге ~/contrib/fs/:

3.1. Файл **hosts_cluster.yml** – пример заполнения файла в режиме отказоустойчивой установки.

3.2. Файл **hosts_sa.yml** – пример заполнения файла для установки в режиме без поддержки отказоустойчивости.

4. Скопировать необходимый файл следующей командой:

```
[root@fs-installer ~]# cp contrib/fs/hosts_sa.yml hosts.yml
```

5. Заполнить файл **inventory**. При этом необходимо заменить параметр **tier** в соответствии с именем пакета установки (fs_client_name, fs_env_name, fs_installation_name). Данный параметр используется для параметров подсистемы объектного хранения данных (в части имени кластера), подсистемы мониторинга инфраструктуры (в части указания окружения), и т.д.

	Параметр **tier** должен соответствовать имени группы в секции 2.

6. Скопировать ssl-ключи для внешнего домена в каталог **certificates**. Подробнее про размещение ключей можно прочитать в п. 3.2.2.3.1.

7. Создать в папке групповых переменных (**group_vars**) каталог для группы серверов с именем, которое было присвоено в файле **inventory**.

Скопировать туда файл с примером заполнения переменных с помощью команды:

```
[root@fs-installer ~]# mkdir group_vars/fs_test_installation/
[root@fs-installer ~]# cp contrib/fs/group_vars/fs_installation/main.yml
group_vars/fs_test_installation/main.yml
```

8. Открыть файл из каталога размещения.
9. Отредактировать значение параметров по комментариям. Примеры параметров для минимальной настройки можно найти в п. 3.2.3.1.

3.2.2.3.1 Размещение ssl-сертификатов для шифрования

Имена сертификатов могут быть произвольными, но они потребуются для дальнейшего заполнения параметров групповых переменных, поэтому важно их запомнить. В документации далее используются примеры имен.

Порядок размещения сертификатов:

1. Разместить сертификат внешнего домена:

```
[root@fs-installer ~]# vim certificates/external_server.cert.pem
```

2. Разместить ключ внешнего домена:

```
[root@fs-installer ~]# vim certificates/external_server.key.pem
```

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

```
[root@fs-installer ~]# vim certificates/external_server.ca.pem
```

В конце файла не должно быть пустой строки. Рекомендуется прочитать файл с помощью утилиты CAT. В результате отобразится следующее:

```
[root@fs-installer ~]# cat certificates/external_server.cert.pem
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
[root@fs-installer ~]#
```

На следующем шаге при заполнении переменных с использованием тестовых имен файлов получатся переменные вида, приведенного в Таблице 9.

Таблица 9 – Пример заполнения переменных

```
setup:
  tls:
    cert_filename: "external_server.cert.pem"
    key_filename: "external_server.key.pem"
    ca_filename: "external_server.ca.pem"
```

3.2.3 Настройка основных параметров установки

3.2.3.1 Минимальные параметры установки

Минимальные параметры, обязательные для заполнения:

- **ansible_user;**
- **setup;**
- **docker;**
- **iptables;**
- **keepalived;**
- **unbound;**
- **etcd;**
- **stolon;**
- **redis;**
- **swift;**
- **postfix;**
- **fs.**

Структура, а также способы заполнения указанных параметров, приведены в таблицах ниже.

3.2.3.1.1 Настройка параметров ОС

	Все примеры для установки в режиме без поддержки отказоустойчивости предоставляются в целях демонстрации функциональности Хранилища. Поддержка данного режима не осуществляется, не рекомендуется проводить такую установку.

Настройка параметров ОС приведена в таблице 10.

Таблица 10 – Настройка параметров ОС

Параметр	Описание	Тип	Значение по умолчанию
ansible_user:	Указывается имя пользователя, имеющего доступ по **ssh** на целевые сервера с машины оператора	Str	admin
iptables:	Параметр настройки управление конфигурациями брандмауэра		
enabled:	Включение автоматической настройки службы управления межсетевым экраном. Параметр рекомендуется оставить включенным (True)	Bool	True

Примеры корректно настроенных параметров:

```
ansible_user: "admin"
iptables:
  enabled: True
```

3.2.3.1.2 Настройка параметров установки (setup)

Настройка данных параметров приведена в Таблице 11.

Таблица 11 – Настройка параметров установки (**setup**)

Параметр		Описание	Тип	Значение по умолчанию
setup:		Параметр настройки установки		
	default_installation_domain	Указывается внешнее доменное имя инсталляции	Str	
	infra_hostname:	Указывается адрес сервера инфраструктурной роли	Str	
	release:	Указывается имя релиза. Например: если имя релиза MyOffice_FS_2019.03, тогда релиз будет 2019.03	Str	

Примеры корректно настроенных параметров:

```
defaul_installation_domain: "mydomain.org"
infra_hostname: "infra01.installation.example.net"
release: "2019.02.21"
```

3.2.3.1.3 Настройка параметров TLS

Настройка данных параметров приведена в Таблице 12.

Таблица 12 – Настройка параметров **TLS**

Параметр		Описание	Тип	Значение по умолчанию
TLS:		Параметр настройки **TLS**		
	cert_filename:	Указывается имя файла сертификата домена		
	key_filename:	Указывается имя файла ключа для сертификата домена		
	ca_filename:	Указывается имя файла сертификата CA		

Примеры корректно настроенных параметров:

```
cert_filename: "external.example.net.pem"
key_filename: "external.example.net-key.pem"
ca_filename: "external.example.net-ca.pem"
```

3.2.3.1.4 Настройка параметров Docker

Настройка параметров приведена в таблице 13.

Таблица 13 – Настройка параметров **Docker**

Параметр	Описание	Тип	Значение по умолчанию
docker:	Блок параметров настройки **Docker**		
image_tag:	Данный параметр не редактируется. В нем используются значения, указанные для параметра setup (см. 3.2.3.1.2)		
registry:	Параметр настройки хранилища образов **Docker**		
endpoint:	Данный параметр не редактируется. В нем используются значения, указанные для параметра setup (см. 3.2.3.1.2)		
daemon:	Блок параметров настройки демона **Docker**		
parameters:	Указываются любые валидные параметры для службы в виде словаря. Параметры будут сконвертированы в json и записаны в файл daemon.json для всех машин. Подробнее см. https://docs.docker.com/engine/reference/commandline/dockerd/		
"insecure-registries":	Заполняется автоматически	list[str]	
"dns":	Указывается список dns-серверов для контейнеров. Рекомендуется использовать **unbound**	list[str]	

		(устанавливается на сервера с ролью **fs_db** или **standalone**)		

Пример корректно настроенных параметров:

```
docker:
  daemon:
    parameters:
      "dns":
        # db01.example.org
        - "192.168.1.8"
        # db02.example.org
        - "192.168.1.9"
        # db03.example.org
        - "192.168.1.10"
```

3.2.3.1.5 Настройка параметров пакетного репозитория (package_tools)

	Данный параметр не редактируется. В нем используются значения, указанные для параметра **setup** (см. 3.2.3.1.2).

3.2.3.1.6 Настройка параметра подсистемы высокой доступности (keepalived)

	Данный параметр не настраивается при проведении установки без поддержки отказоустойчивости. Для его отключения потребуется прописать: "fs.cluster: False".

Настройка параметра приведена в Таблице 14.

Таблица 14 – Настройка параметра подсистемы высокой доступности (**keepalived**)

Параметр	Описание	Тип	Значение по умолчанию
keepalived:	Блок параметров настройки подсистемы высокой доступности		
vrrp:	Блок настроек протокола обеспечения отказоустойчивости шлюза		
instances:	Блок настроек ролей		
fs_fe:	Настройка виртуального ip-адреса для FSApi		
virtual_ip:	Указывается виртуальный ip-адрес		
password:	Указывается пароль		
fs_db:	Настройка виртуального ip-адреса для БД PostgreSQL		
virtual_ip:	Указывается виртуальный ip-адрес		
password:	Указывается пароль		

Пример корректно настроенных параметров:

```
keepalived:
  vrrp:
    instances:
      fs_fe:
```

```
    virtual_ip: "191.238.74.148 "
    password: "x14Thn20q"
  fs_db:
    virtual_ip: "191.238.74.148 "
    password: "x14Thn20q"
```

3.2.3.1.7 Настройка параметров кеширующего DNS-сервера (unbound)

	Данный параметр не рекомендуется отключать! Отключение параметра может привести к ухудшениям в работе Хранилища.

Таблица 15 – Настройка параметров кеширующего DNS-сервера (**unbound**)

Параметр	Описание	Тип	Значение по умолчанию
unbound:	Блок параметров настройки кеширующего DNS-сервера		
access_control:	Блок настройки адреса для доступа к сети **Docker** (предварительно настроен, не рекомендуется вносить изменения)		
- "172.17.0.0/16"	Указывается список подсетей, которым будет разрешено обращаться к **unbound** за разрешением имен		
local_zone:	Блок настройки часового пояса		
- type:	Указывается значение **transparent**, **nodefault** или **redirect**		
zone:	Используется по умолчанию как домен для установки		
local_data:	Указываются записи для каждого сервера в группе **fs** и отдельное доменное имя для **etcd**		
domain:	Указывается домен		
type:	Указывается значение **transparent**, **nodefault** или **redirect**		

	ip:	Указывается ip-адрес		

Пример корректно настроенных параметров:

```
unbound:
  access_control:
    - "172.17.0.0/16"
  local_zone:
    - type: "transparent""
      zone: "cluster.example.net"
      local_data:
# Пример заполнения параметров в режиме отказоустойчивой установки
        - domain: "be01.installation.example.net"
          type: "A"
          ip: "192.168.1.2"
        - domain: "be02.installation.example.net"
          type: "A"
          ip: "192.168.1.3"
        - domain: "fe01.installation.example.net"
          type: "A"
          ip: "192.168.1.4"
        - domain: "fe02.installation.example.net"
          type: "A"
          ip: "192.168.1.5"
        - domain: "st01.installation.example.net"
          type: "A"
          ip: "192.168.1.6"
        - domain: "st02.installation.example.net"
          type: "A"
          ip: "192.168.1.7"
        - domain: "st03.installation.example.net"
          type: "A"
          ip: "192.168.1.8"
        - domain: "db01.installation.example.net"
          type: "A"
          ip: "192.168.1.9"
        - domain: "db02.installation.example.net"
          type: "A"
          ip: "192.168.1.10"
```

```
    - domain: "db03.installation.example.net"
      type: "A"
      ip: "192.168.1.11"
    - domain: "etcd.db01.installation.example.net"
      type: "A"
      ip: "192.168.1.9"
    - domain: "etcd.db02.installation.example.net"
      type: "A"
      ip: "192.168.1.10"
    - domain: "etcd.db03.installation.example.net"
      type: "A"
      ip: "192.168.1.11"
    - domain: "infra01.installation.example.net"
      type: "A"
      ip: "192.168.1.12"
# Пример заполнения параметров в режиме без поддержки отказоустойчивости
    - domain: "sa01.installation.example.net"
      type: "A"
      ip: "192.168.1.2"
    - domain: "etcd.sa01.installation.example.net"
      type: "A"
      ip: "192.168.1.2"
```

3.2.3.1.8 Настройка хранилищ

Для корректной установки ПО требуется настроить хранилища **ETCD**, **Stolon**, **Redis** и **Swift**.

Таблица 16 – Настройка параметров хранилищ

Параметр	Описание	Тип	Значение по умолчанию
ETCD:	Блок параметров настройки отказоустойчивого хранилища		
token:	Указывается разделяемый секрет между экземплярами кластера, требуется для установки соединения. Например: «fs_42_3»		
cluster:	Указывается список серверов, на которых будут установлены хранилища **ETCD**		
Stolon:	Блок настройки СУБД PostgreSQL		

	postgres_password:	Указывается пароль пользователя **postgres. Не может содержать специальных символов!**		
	replication:	Параметр настройки дубликата БД		
	password:	Указывается пароль пользователя **replication**		
	monitoring:	Параметр настройки мониторинга		
	password:	Указывается пароль пользователя **monitoring**		
	users:	Параметры настройки пользователей		
	fs:	Параметры настройки **fs**		
	password:	Указывается пароль для базы данных **fs**. Генерируется с помощью команды pwgen 32		
	fs_search:	Параметры настройки **fs_search**		
	password:	Указывается пароль для базы данных поиска **fs**. Генерируется с помощью команды pwgen 32		
	pg_parameters:[6]	Блок настройки параметров PostgreSQL – оптимизация производительности. Требуется указывать такие единицы измерения: kB, MB, GB либо TB		
	max_connections:	Указывается количество максимальных подключений		
	shared_buffers:	Указывается объем памяти, который будет использовать сервер БД для буферов в разделяемой памяти		
	effective_cache_size:	Указывается допустимый размер кеша для одного запроса		4 GB
	maintenance_work_mem:	Указывается максимальный объем памяти для операций обслуживания БД		64 MB
	checkpoint_completion_target:	Указывается целевое время для завершения		0.5

[6] Далее приведен пример заполнения параметров, оптимизированный под 16 GB. Для расчета параметров под другой объем памяти необходимо использовать https://pgtune.leopard.in.ua/.

		процедуры контрольной точки		
	wal_buffers:	Указывается объем разделяемой памяти, который будет использоваться для буферизации данных WAL, еще не записанных на диск		
	default_statistics_target:	Устанавливается целевое ограничение статистики по умолчанию, распространяющееся на столбцы, для которых командой ALTER TABLE SET STATISTICS не заданы отдельные ограничения		100
	random_page_cost:	Указывается приблизительная стоимость чтения одной произвольной страницы с диска		4.0
	effective_io_concurrency:	Указывается допустимое число параллельных операций ввода/вывода		
	work_mem:	Указывается объем памяти, который будет использоваться для внутренних операций сортировки и хеш-таблиц, прежде чем будут задействованы временные файлы на диске		4 MB
	min_wal_size:	Указывается минимальный размер, до которого может		80 MB

		вырастать WAL между автоматическими контрольными точками в WAL		
	max_wal_size:	Указывается максимальный размер, до которого может вырастать WAL между автоматическими контрольными точками в WAL		1 GB
	max_worker_processes:	Указывается максимальное число фоновых процессов, которое можно запустить в Хранилище		8
Redis:		Блок настроек хранилища **Redis**		
	cluster:	Параметр настройки кластера		
	master:	Параметр настройки кластера **master**		
	node_number:	Указывается номер кластера **master** хранилища **Redis**, например, "01"		
	hosting_server:	Указывается **Hostname** сервера, на котором будет размещено хранилище **Redis**. Параметр должен соответствовать значению **hostname** в файле **inventory**		
	password:	Указывается пароль для доступа к хранилищу **Redis**		
	slaves:	Указывается список узлов		

		сети **slave**[7]		
	- node_number:	Указывается номер узла сети **slave** хранилища **Redis**, например, "02"		
	hosting_server:	Указывается **Hostname** сервера, на котором будет размещено хранилище **Redis**. Параметр должен соответствовать значению **hostname** в файле **inventory**		
sentinel:		Блок настроек хранилища с подсистемой отказоустойчивости **Redis-sentinel**		
	cluster:	Указывается список узлов кластера промежуточной среды		
	- node_number:	Указывается номер узла сети кластера, например, "01"		
	hosting_server:	Указывается **Hostname** сервера, на котором будет размещен кластер		
Swift:		Блок настроек подсистемы объектного хранения данных		
cluster:		Параметр настройки кластера		
	nodes:	Указывается список узлов кластера		
	- node_number:	Указывается номер узла кластера. Например, "01"		
	hosting_server:	Указывается **Hostname** сервера, на котором будет размещен узел. Параметр должен соответствовать **hostname** в файле **inventory**		
	drives:	Указывается список дисков для хранения		

[7] Только для установки в отказоустойчивом режиме.

		данных. Диск должен быть примонтирован в раздел /srv/node/<drive_name>		
	Swift: users: fs: key	Указывается пароль для **Swift**. Требуется сгенерировать с помощью команды pwgen 32		

Пример корректно настроенных параметров:

```
etcd:
  token: "fs_42_3"
  cluster:
# Пример заполнения параметров в режиме отказоустойчивой установки
    - "etcd.db01.installation.example.net"
    - "etcd.db02.installation.example.net"
    - "etcd.db03.installation.example.net"
# Пример заполнения параметров  в режиме без поддержки отказоустойчивости
    - "etcd.sa01.installation.example.net"
stolon:
  replication:
    password: "x14Thn20q"
  monitoring:
    password: "x14Thn20q"
# Пример заполнения пароля пользователя
  users:
    fs:
     key: "achoh5iju7Teith6aibomahl7oj2Ai6o"
# Пример заполнения ниже рассчитан на 16 GB RAM!
  pg_parameters:
    max_connections: "128"
    shared_buffers: "4GB"
    effective_cache_size: "12GB"
    maintenance_work_mem: "1GB"
    checkpoint_completion_target: "0.7"
    wal_buffers: "16MB"
```

```
    default_statistics_target: "100"
    random_page_cost: "4"
    effective_io_concurrency: "2"
    work_mem: "8MB"
    min_wal_size: "1GB"
    max_wal_size: "2GB"
    max_worker_processes: "8"
redis:
  cluster:
# Пример заполнения параметров в режиме без поддержки отказоустойчивости
    master:
      node_number: "01"
      hosting_server: "db01.installation.example.net"
      password: "x14Thn20q"
# Пример заполнения параметров в режиме отказоустойчивой установки
    slaves:
      - node_number: "02"
        hosting_server: "db02.installation.example.net"
      - node_number: "03"
        hosting_server: "db03.installation.example.net"
  sentinel:
# Пример заполнения параметров при в режиме без поддержки отказоустойчивости
    cluster:
      - node_number: "01"
        hosting_server: "db01.installation.example.net"
# Пример заполнения параметров в режиме отказоустойчивой установки
      - node_number: "02"
        hosting_server: "db02.installation.example.net"
      - node_number: "03"
        hosting_server: "db03.installation.example.net"
swift:
  cluster:
# Пример заполнения параметров в режиме без поддержки отказоустойчивости
    nodes:
      - node_number: "01"
        hosting_server: "db01.installation.example.net"
```

```
      drives:
# Диск /srv/node/data0
        - "data0"

# Пример заполнения параметров в режиме отказоустойчивой установки
    - node_number: "02"
      hosting_server: "st02.installation.example.net"
      drives:
# Диск /srv/node/data0
        - "data0"
    - node_number: "03"
      hosting_server: "st03.installation.example.net"
      drives:

# Диск /srv/node/data0
        - "data0"
```

3.2.3.1.9 Настройка параметров почтового сервера (postfix)

Настройка параметров почтового сервера (**postfix**) приведена в Таблице 17.

Таблица 17 – Настройка параметров почтового сервера (**postfix**)

Параметр	Описание	Тип	Значение по умолчанию
postfix:	Параметр настройки почтового сервера		
relay:	Параметр настройки почтовой передачи		
relayhost:	Указывается **hostname** почтовой передачи (PSN)		psn-sa-1.installation.example.net или smtp.example.net
name:	Указывается имя почтовой передачи		
parameters:	Блок настройки параметров почтовой передачи		
default:	Указывается домен, используемый по умолчанию		

	mydomain:	Указывается домен, для которого осуществляется доставка почты в инсталляции		

Пример корректно настроенных параметров:

```
postfix:
  relay:
    relayhost:
      name: "psn-sa-1.installation.example.net"
    parameters:
      default:
        mydomain: "cluster.example.net"
```

3.2.3.1.10 Настройка параметров Хранилища (fs)

	При миграции с предыдущих релизов Хранилища требуется использовать те же значения параметров **salt1**-**salt6**, **salt_no_captcha** и **salt_no_auth**, что и использовались ранее. В противном случае пользователи установки не смогут авторизоваться При установке в режиме без поддержки отказоустойчивости задается параметр **cluster**: False.

Настройка параметров Хранилища приведена в Таблице 18

Таблица 18 – Настройка параметров Хранилища

Параметр	Описание	Тип	Значение по умолчанию
fs:	Блок параметров настройки Хранилища		
cluster:	Параметр настройки отказоустойчивого кластера	bool	True
server_id:	Указывается ID сервера (номер установки): **не более 5 цифр!**	str	"777"[8]
default_domain:	Указывается домен, который по умолчанию используется для установки	str	
domain_module:	Задается шаблон для генерирования URL "{service}.{domain}" для URL вида "auth.stageoffice.ru" "{service}-{domain}" для URL вида "auth-stageoffice.ru" (где stageoffice.ru – fs.default_domain)	str	
allowed_domains:	Указывается список разрешенных доменов	list[str]	
esia:	Параметр настройки взаимодействия с ЕСИА		
enabled:	Параметр включения взаимодействия	bool	False
psn:	Параметр настройки взаимодействия с PSN		
enabled:	Параметр включения взаимодействия	bool	False
kerberos:	Параметр настройки взаимодействия с Active Directory		
enabled:	Параметр включения взаимодействия	bool	False
default_realm:	Параметр настройки	str	**Kerberos Realm**

[8] Значение, указываемое для **server_id**, обязательно заключается в "" (кавычки)!

		области (**realm**) по умолчанию		
	realms:	Указывается список **Kerberos Realms**	list[dict]	
	- name:	Указывается имя домена	str	
	controllers:	Указывается список контроллеров домена	list[str]	
tenants:		Параметр настройки тенантов		
	default:	Параметр настройки тенанта по умолчанию		
	default_domain:	Указывается имя домена	str	
	recovery_email:	Указывается электронный почтовый адрес для восстановления пароля	str	
	admin:	Параметр настройки аккаунта администратора		
	password:	Указывается пароль администратора	str	
ssl:		Параметр настройки сертификатов для HTTPS		
	service:	Параметр настройки сервиса		
	cert_filename:	Указывается имя сертификата	str	
	key_filename:	Указывается имя ключа	str	
salts:		Блок параметров настройки солей шифрования		
	shared:	Указывается соль шифрования внутренних запросов. Генерируется при помощи команды pwgen –s 16	str	
	salt1:	Указывается случайная соль шифрования сессий и токенов. Генерируется при помощи команды pwgen –s 16.	str	
	salt2:		str	
	salt3:		str	
	salt4:		str	
	salt5:		str	
	salt6:		str	
	salt_no_captcha:	Указывается параметр CO	str	

		FS_TOKEN_SALT		
	salt_no_auth:	Указывается параметр CO FS_TOKEN_SALT_EXT	str	
crypt:[9]		Блок параметров настройки криптографических ключей (обязательный словарь для заполнения)		
	main:	Блок для заполнения **aes_iv**, **aes_key**, **aes_salt**		
	aes_iv:	Указывается параметр CO FS_APP_ENCRYPTION_IV	str	
	aes_key:	Указывается параметр CO FS_APP_ENCRYPTION_KEY	str	
	aes_salt:	Указывается параметр CO AUTH_ENCRYPTION_SALT	str	
	session:	Параметр настройки сессии		
	aes_iv:	Указывается параметр CO AUTH_ENCRYPTION_IV	str	
	aes_key:	Указывается параметр CO AUTH_ENCRYPTION_KEY	str	
	aes_salt:	Указывается параметр CO AUTH_ENCRYPTION_SALT	str	
mtadminapi_user:		Параметр настройки доступа к **MTAdminAPI**		
	password:	Указывается пароль администратора	str	
appapi:		Параметр настройки **AppAPI**	Конфигурация AppAPI	
	log:	Параметр настройки логирования		
	level:	Указывается уровень логирования	str	
	accounts:	Параметр настройки APP-учетных записей		
	co:	Параметр настройки учетной записи **app-co**		
	password:	Указывается пароль к учетной записи **app-co**	str	
	psn:	Параметр настройки учетной записи **app-psn**		
	password:	Указывается пароль к	str	

[9] Генерация всех значений секций **main** и **session** словаря **crypt** выполняется при помощи команды: openssl enc -aes-256-cbc -k "<pass_phrase>" -P -md sha256. Если словарь **crypt** не заполнен, то аутентификация не будет работать.

		учетной записи **app-psn**		
	logos:	Параметр настройки учетной записи **app-logos**		
	password:	Указывается пароль к учетной записи **app-logos**	str	
password_policy:		Параметр настройки парольной политики		
	min_length	Указывается минимальная длина пароля	int	8
	min_numbers	Указывается минимальное количество цифр в пароле	int	1
	min_special_symbols	Указывается минимальное количество специальных символов в пароле	int	0
	letters_required	Включается требование наличия букв в пароле	bool	True
	mixed_case_required	Включается требование наличия букв в разном регистре		False
	default_expire_time	Указывается время истечения срока действия пароля	int	0
captcha:		Параметр настройки капчи		
	fail_captcha_start	Указывается количество попыток неправильного ввода пароля для включения капчи	int	3
	fail_ip_skip	Указывается белый лист для IP-адреса клиента	str	"10.20."
	fail_lifetime	Указывается время, за которое клиенту будет требоваться ввести капчу	int	3600
	fail_max_attempts	Указывается максимальное количество попыток неправильного ввода пароля для отключения авторизации	int	8
	fail_timeout	Указывается время, на которое отключается	int	300

		авторизация		
	ratelimit:	Параметр настройки ограничения скорости		
	enabled	Включается ограничение скорости	bool	True
	bruteforce_attempts	Указывается количество попыток подбора пароля для включения капчи	int	3000
	bruteforce_period	Указывается период, в который должно состояться включение капчи	int	1000
	ddos_attempts	Указывается количество запросов с одного адреса для включения капчи	int	6000
	ddos_period	Указывается период, в который должно состояться включение капчи	int	3000

Пример корректно настроенных параметров:

```
fs:
  # При установке без поддержки отказоустойчивости выставить значение False
  cluster: False
  server_id: "1"

  default_domain: "example.com"
  domain_module: "{service}-{domain}"
  allowed_domains: ["example.com"]

  esia:
    enabled: False

  psn:
    enabled: False

  kerberos:
    enabled: False
    default_realm: "co.com"
    realms:
```

```
    - name: "co.com"
      controllers:
        - "dc1.co.com"
        - "dc2.co.com"

tenants:
  default:
    default_domain: "example.com"
    recovery_email: "admin@example.com"
    admin:
      password: "password"

ssl:
  service:
    cert_filename: "bundle_"
    key_filename:

salts:
  shared: Aed4zoh6neiSoephoh8sho5boh0xei8t
  salt1: aeshahXi6Roosao5goLu9heesieshahk
  salt2: aephie3pheel9Ahtho7uyiga0quufaiC
  salt3: aevaec0eth9AebaeCahlaiSie7uGaphu
  salt4: oochai0siid5Ieghah5boh5ohfooBooV
  salt5: feop1rohSe7ea6Chooc8aeZ4rahT8yee
  salt6: othoochae2ohnguKoy6pha1Ekei7aiYi
  salt_no_captcha: uCaquaezooh2ju6Ouy2Choco4paipood
  salt_no_auth: weih2eem9AiH5nie1poh3Iedahdig6sh


crypt:
  main:
    aes_iv: 824C43CE2537FE1ED385D4175E56D313
    aes_key: 1E16426C92D394737849C262F2DDD5EF88117204A4B451BFACC709D30D22C05B
    aes_salt: 8B5B3D8B57FC9EDF

  session:
    aes_iv: 0DAE509ABDD010BEBACC967D744B6ADB
    aes_key: 69A88F9F4CC1F579DA3803383F4101440AAED12325E15EA74FE6256FB175A5F9
```

```
    aes_salt: 7AFC6350E921C247

mtadminapi_user:
  password: "password"

appapi:
  log:
    level: "info"
  accounts:
    co:
      password: "changeme12345"
    psn:
      password: "changeme12345"
    logos:
      password: "changeme12345"

password_policy:
  min_length: 8
  min_numbers: 1
  min_special_symbols: 0
  letters_required: True
  mixed_case_required: False
  default_expire_time: 0

captcha:
  fail_captcha_start: 3
  fail_ip_skip: "10.20."
  fail_lifetime: 3600
  fail_max_attempts: 8
  fail_timeout: 300

ratelimit:
  enabled: True
  bruteforce_attempts: 3000
  bruteforce_period: 1000
  ddos_attempts: 6000
  ddos_period: 3000
```

3.2.3.2 Настройка без отказоустойчивости

	Не рекомендуется производить настройку и выполнять установку Хранилища на оборудовании без поддержки отказоустойчивости. Программное обеспечение предназначается только для отказоустойчивого оборудования (поддержка от 11 серверов).

При выполнении такой установки дополнительно потребуется заполнить параметры, указанные в таблице 19.

Таблица 19 – Заполнение параметров в режиме без поддержки отказоустойчивости

Для параметра FS:	**Параметр**	**Пример заполнения**
	cluster:	False
Для параметра unbound:	local_data:	Указываются сведения о доменах, типе и ip. Например: • domain: "sa01.installation.example.net"; • type: "A"; • ip: "192.168.1.2"; • domain: "etcd.sa01.installation.example.net"; • type: "A"; • ip: "192.168.1.3"
Для параметра docker:	dns:	Пример заполнения: sa01.installation.example.net – "192.168.1.2"

3.2.3.3 Отказоустойчивая установка

При выполнении такой установки дополнительно потребуется заполнить параметры, указанные в таблице 20.

Таблица 20 – Заполнение параметров в отказоустойчивом режиме

Для параметра Swift:	**Описание**	**Значение по умолчанию**
	node_number:	02
	hosting_server:	st02.installation.example.net
	drivers:	data0
	node_number:	03
	hosting_server:	st03.installation.example.net
	drivers:	data0
Для параметра Redis:	slaves:	кластер
	node_number:	02
	hosting_server:	db02.installation.example.net
	node_number:	03
	hosting_server:	db03.installation.example.net
	node_number:	02
	hosting_server:	db02.installation.example.net
	node_number:	03
	hosting_server:	db03.installation.example.net
	tunnel:	<не заполняется>
	nodes:	<не заполняется>
	node_number:	01
	hosting_server:	be01.installation.example.net
	databases:	<не заполняется>
	name:	fs_master
	local_port:	6379
	node_number:	02
	hosting_server:	e02.installation.example.net
	databases:	<не заполняется>
	name:	fs_master

	local_port:	6379
Для параметра unbound:	local_data:	Указываются сведения о доменах, типе и ip. Например: • domain: "be01.installation.example.net"; • type: "A"; • ip: "192.168.1.2"; • domain: "be02.installation.example.net"; • type: "A"; • ip: "192.168.1.3"; • domain: "fe01.installation.example.net"; • type: "A"; • ip: "192.168.1.4"; • domain: "fe02.installation.example.net"; • type: "A"; • ip: "192.168.1.5"; • domain: "st01.installation.example.net"; • type: "A"; • ip: "192.168.1.6"; • domain: "st02.installation.example.net"; • type: "A"; • ip: "192.168.1.7"; • domain: "st03.installation.example.net"; • type: "A"; • ip: "192.168.1.8"; • domain: "db01.installation.example.net"; • type: "A"; • ip: "192.168.1.9"; • domain: "db02.installation.example.net"; • type: "A";

		• ip: "192.168.1.10"; • domain: "db03.installation.example.net"; • type: "A"; • ip: "192.168.1.11"; • domain: "etcd.db01.installation.example.net"; • type: "A"; • ip: "192.168.1.9"; • domain: "etcd.db02.installation.example.net"; • type: "A"; • ip: "192.168.1.10"; • domain: "etcd.db03.installation.example.net"; • type: "A"; • ip: "192.168.1.11"; • domain: "infra01.installation.example.net"; • type: "A"
Для параметра iptables:	keepalived:	True или False
Для параметра docker:	dns:	Пример заполнения: db01.installation.example.net – "192.168.1.9"; db02.installation.example.net – "192.168.1.10"; db03.installation.example.net – "192.168.1.11"

3.2.4 Настройка дополнительных параметров установки

Примеры заполнения расширенных параметров установки Хранилища приведены в Приложении Г настоящего руководства.

3.2.5 Настройка межсетевого экранирования

	Во время установки на все сервера будет установлена служба управления межсетевым экраном **iptables** и настроены правила, ограничивающие входящий доступ по всем портам, кроме тех, которые занимают запущенные контейнеры на соответствующих серверах, и разрешены заданными правилами экрана

Сетевые порты, используемые контейнерами, а также установленные правила межсетевого экрана приведены в таблицах ниже.

Таблица 21 – Сетевые порты

	Связанный IP	**Порт**	**Протокол**	**Назначение**
Сервера роли fs_fe	0.0.0.0	80	TCP	HTTP-доступ к Хранилищу и API
	172.17.0.1	81		Мониторинг веб-сервера (**nginx**)
	0.0.0.0	443		HTTPS-доступ к Хранилищу и API
Сервера роли fs_be	172.17.0.1	26		SMTP для внутренних служб (выполняет перенаправление на PSN)
	172.17.0.1	5432		Прокси подсистемы отказоустойчивости службы управления базами данных (**stolon**)
	0.0.0.0	8080		HTTPS-доступ к Хранилищу и API
	172.17.0.1	8280		Прокси подсистемы объектного хранения данных (**swift**)
	0.0.0.0	9090		Управление почтовым сервисом (**mtadmin**)
	0.0.0.0	9091		Управление API Хранилища (веб-версия)
	0.0.0.0	9092		Управление API (веб-версия)
	0.0.0.0	9095		Модуль **MailAPI**, предназначенный для управления пользовательской адресной книгой (**CardAPI**)
	0.0.0.0	17777		Прокси Хранилища (**fs**)
Сервера роли fs_db	0.0.0.0	2379		Клиентский порт отказоустойчивого хранилища (**ETCD**)
	0.0.0.0	2380		Кластерный порт отказоустойчивого хранилища (**ETCD**)
	0.0.0.0	5432		Прокси подсистемы отказоустойчивости СУБД (**Stolon**)

	0.0.0.0	6379		Клиентский порт хранилища (**Redis**)
	0.0.0.0	16379		Кластерный порт хранилища (**Redis**)
	0.0.0.0	25432		Управление экземплярами PostgreSQL (**stolon keeper**)
	0.0.0.0	26379		Подсистема отказоустойчивости хранилища **Redis**
Сервера роли fs_st	0.0.0.0	873		Служба синхронизации файлов подсистемы объектного хранения данных
	0.0.0.0	6000		Подсистема объектного хранения данных
	0.0.0.0	6001		Сервер контейнеров подсистемы объектного хранения данных
	0.0.0.0	6002		Сервер аккаунтов подсистемы объектного хранения данных
	0.0.0.0	8280		Прокси-сервер подсистемы объектного хранения данных
	0.0.0.0	11211		Сервис кеширования данных (**memcached**)
Сервера роли fs_infra	0.0.0.0	5000		Хранилище образов **Docker** (**docker registry**)
	0.0.0.0	8081		rpm-репозиторий веб-сервера (**nginx**)

Таблица 22 – Установленные правила межсетевого экрана

	Таблица	Состояние	Порт	Протокол	Интерфейс	Разрешение
Сервера роли fs_fe	INPUT					DROP
	FORWARD					DROP
	OUTPUT					ACCEPT
	INPUT	RELATED, ESTABLISHED				ACCEPT
	INPUT			ICMP		ACCEPT
	INPUT				lo	ACCEPT
	INPUT	NEW	SSH	TCP		ACCEPT
	INPUT			VRRP		ACCEPT
Сервера роли fs_be	INPUT					DROP
	FORWARD					DROP
	OUTPUT					ACCEPT
	INPUT	RELATED, ESTABLISHED				ACCEPT
	INPUT			ICMP		ACCEPT
	INPUT				lo	ACCEPT
	INPUT	NEW	SSH	TCP		ACCEPT
	INPUT				docker0	ACCEPT
Сервера роли fs_db	INPUT					DROP
	FORWARD					DROP
	OUTPUT					ACCEPT
	INPUT	RELATED, ESTABLISHED				ACCEPT
	INPUT			ICMP		ACCEPT
	INPUT				lo	ACCEPT
	INPUT	NEW	SSH	TCP		ACCEPT
	INPUT			VRRP		ACCEPT
					docker0	ACCEPT
		NEW	53	UDP		ACCEPT
Сервера роли fs_st	INPUT					DROP
	FORWARD					DROP
	OUTPUT					ACCEPT
	INPUT	RELATED,				ACCEPT

		ESTABLISHED				
	INPUT			ICMP		ACCEPT
	INPUT				lo	ACCEPT
	INPUT	NEW	SSH	TCP		ACCEPT
	INPUT				docker0	ACCEPT
Сервера роли fs_infra	INPUT					DROP
	FORWARD					DROP
	OUTPUT					ACCEPT
	INPUT	RELATED,ESTABLISHED				ACCEPT
	INPUT			ICMP		ACCEPT
	INPUT				lo	ACCEPT
	INPUT	NEW	SSH	TCP		ACCEPT
	INPUT	NEW	5000	TCP		ACCEPT
	INPUT	NEW	8081	TCP		ACCEPT

3.2.6 Настройка удаленного доступа

Настройка удаленного доступа выполняется при помощи роли **sshd**. Пример настройки роли приведен ниже:

```
ansible_port: 22
sshd:
  protocol: 2
  accept_env: "LC_*"
  permit_root_login: "no"
  password_authentication: "yes"
  use_dns: "no"
  x11_forwarding: "no"
  allow_groups: []
   allow_users: []
```

3.3 Установка Хранилища

3.3.1 Конфигурация без отказоустойчивости

3.3.1.1 Запуск установки

Для установки потребуется совершить следующие действия:

1. Запустить команду на подготовку серверов к установке:

```
[root@fs-installer ~]# ansible-playbook playbooks/common.yml --diff
```

После запуска этой команды будут запущены роли, указанные в таблице 1.

2. Запустить команду на установку Хранилища:

```
[root@fs-installer ~]# ansible-playbook playbooks/FS/main.yml --diff
```

После этого запускаются следующие роли:

- **check_installation** – проверка всех требуемых переменных для начала установки;
- **keepalived** на **fs_db** группу – установка и настройка подсистемы высокой доступности (только в случае установки на кластер);
- **unbound** – установка и настройка кеширующего DNS (ускоряет работу внутри стенда установки, создает возможность работы на серверных IP);
- **ETCD** – установка и настройка распределенного хранилища (**key-value**);
- **Stolon** на **fs_db** группу – установка и настройка СУБД Postgres с подсистемой отказоустойчивости **stolon**;
- **Redis** на **fs_db** группу – установка и настройка хранилища структуры данных **Redis** с подсистемой отказоустойчивости **sentinel**;
- **Swift** – установка и настройка подсистемы объектного хранения данных;
- **Stolon** на **fs_be** группу – установка и настройка прокси до СУБД Postgres с подсистемой отказоустойчивости **stolon**;
- **Redis** на **fs_be** группу – установка и настройка прокси **Redis** с подсистемой отказоустойчивости **sentinel**;
- **postfix** – установка и настройка почтового сервиса (**MTA**);
- **fleet** – установка и настройка менеджера сервисов (**systemd**), связанного с хранилищем **ETCD** в распределенную систему инициализации;
- **fs** (**tag fs_setup**) – установка и настройка всех требуемых сервисов Хранилища;

- **keepalived** на **fs_fe** группу – установка и настройка сервиса высокой доступности (только в случае установки на кластер);
- **nginx** – установка и настройка веб-сервера (**nginx**);
- **confd** – установка и настройка менеджера конфигураций;
- **fs** (**tag fs_accounts**) – создание учетных записей co, psn, logos для приложений.

3.3.1.2 Проверка корректности установки

Для проверки установки Хранилища:

1. Зайти на адрес https://admin-<default_domain>.
2. Проверить, что страница открывается и происходит автоматическая авторизация под пользователем fsadmin@<defaul_domain> и под паролем, указанным в переменных fs.tenants.default.admin.password.

3.3.2 Кластерная отказоустойчивая конфигурация

3.3.2.1 Запуск установки

Описание запуска установки приведено в п. 3.3.1.1.

3.3.2.2 Проверка корректности установки

Описание проверки корректности установки приведено в п. 3.3.1.2.

3.4 Установка в составе МойОфис Частное облако

В случае установки продукта МойОфис Хранилище в составе комплексного продукта МойОфис Частное Облако, продукт Файловое хранилище должен устанавливаться первым.

4 Обновление с предыдущих версий

Данный дистрибутив предназначен для чистой установки либо для миграции с версии 2018.02 Mint. Информация по чистой установке приведена в разделе 3. Информация по миграции описана в настоящем разделе.

	Данный дистрибутив подходит для чистой установки и миграции с версии 2018.02 Mint.

Для миграции данных необходимо сделать резервные копии трех сущностей:

- База данных СУБД PostgreSQL;
- Контейнеры подсистемы объектного хранения данных;
- Распределенное хранилище **ETCD**.

Кроме того, потребуется сделать резервную копию параметров установки в части солей, паролей и т.д.

	При выполнении резервного копирования рекомендуется выключить сервера роли **backend**. Это исключит попадание новых данных в Хранилище в процессе резервного копирования.

4.1 Состав дистрибутива

Состав дистрибутива приведен в п. 3.1.

4.2 Подготовка к обновлению

4.2.1 Описание ролей

Описание ролей приведено в п. 3.2.1.

4.2.2 Проверка и настройка инфраструктуры установки

Описание проверки и настройки инфраструктуры установки приведено в п. 3.2.2.

4.2.3 Проверка и настройка основных параметров установки

Описание проверки и настройки данных параметров приведено в п. 3.2.3.

4.2.4 Проверка и настройка дополнительных параметров установки

Описание проверки и настройки дополнительных параметров установки приведено в п. 3.2.4.

4.2.5 Проверка и настройка межсетевого экранирования

Описание проверки и настроек межсетевого экранирования приведено в п. 3.2.5.

4.2.6 Проверка и настройка удаленного доступа

Описание проверки и настройки удаленного доступа приведено в п. 3.2.6.

4.2.7 Создание резервных копий

4.2.7.1 Создание резервных копий баз данных СУБД PostgreSQL

Создание резервной копии выполняется с помощью утилиты командной строки **pg_dump**. Ее можно использовать для снятия дампа памяти со следующих БД:

- Файловое хранилище;
- Индекс поисковой подсистемы;
- Почта;
- Logos.

4.2.7.1.1 Поиск имен баз данных Файлового хранилища

Точные имена БД можно посмотреть на серверах с ролью **backend** в конфигурационном файле /etc/nct/nct-api.ini.

Ниже приведен пример имен БД, с которых требуется снять резервные копии.

```
pg_search_dbname = fs_search   # БД Индекса поисковой системы
pg_vmail_dbname = fs_vmail     # БД Почты
pg_dbname = fs                 # БД Хранилища
```

Имя БД Logos можно посмотреть на серверах с ролью **logos**, в параметрах запущенного процесса сервиса **baham**. Для вывода параметров в списке процессов необходимо выполнить команду в терминале:

```
[root@lgs1 ~]# ps axu | grep baham | grep -v grep
```

Пример результата вывода после выполнения команды:

```
Messeng+  1366  0.6  0.0  48668 25380 ?        Ssl  фев15 1408:54 /opt/messenger/bin/baham -addr=0.0.0.0:9041 -timeout=30s -from=postgres://fs:password@10.10.10.10/fs?sslmode=disable -to=postgres://lgs:password@10.10.10.10/lgs?sslmode=disable&search_path=catalog,extensions -redis=redis://:password@10.10.10.10 -rabbitmq=amqp://lgs:password@10.10.10.10/lgs,amqp://lgs:lgs@10.10.10.11/lgs -taygeta=:10054 -rigel=:10055 -tls-host-override=logos.nct -ca=/opt/messenger/certs/ca.crt -client-cert=/opt/messenger/certs/client.crt -client-key=/opt/messenger/certs/server.key -json=true -debug=false -statsd=false -vega=https://logos.example.com/vega -login=app-lgs@example.com -pass=password -json=true -debug=false -tls-on
```

В результате нужно найти подстроку:

```
-to=postgres://lgs:password@10.10.10.10/lgs?sslmode=disable&search_path=catalog,extensions
```

Имя БД Logos – **lgs**. Оно располагается после /, закрывающей IP-адрес, и заканчивается до ?.

4.2.7.1.2 Создание резервных копий БД

Убедитесь, что:

- присутствует SSH-доступ на сервер резервного копирования данных;
- создан каталог для сохранения резервной копии;
- каталог доступен выбранному пользователю для записи.

Резервные копии создаются на машинах с работающей СУБД PostgreSQL следующими командами (по команде на каждую БД):

```
[root@db1 ~]# PGPASSWORD="fs_search_password" pg_dump -U db_search_user fs_search | ssh me@backupserver "cat > /backup/fs_oldrelease/fs_search.sql"

[root@db1 ~]# PGPASSWORD="fs_vmail_password" pg_dump -U db_vmail_user fs_vmail | ssh me@backupserver "cat > /backup/fs_oldrelease/fs_vmail.sql"

[root@db1 ~]# PGPASSWORD="fs_password" pg_dump -U db_fs_user fs | ssh me@backupserver "cat > /backup/fs_oldrelease/fs.sql"

[root@db1 ~]# PGPASSWORD="lgs_password" pg_dump -U db_lgs_user lgs | ssh me@backupserver "cat > /backup/fs_oldrelease/lgs.sql"
```

, где:

- PGPASSWORD="fs_search_password" – переменная, содержащая пароль текущего пользователя к выбранной базе;
- **pg_dump** – утилита для создания резервной копии;
- -U fs_search – ключ, указывающий на ввод имени пользователя, далее – само имя пользователя. Данный пользователь имеет полный доступ к соответствующей БД;
- db_search_user – имя целевой БД для создания резервной копии;
- | – перенаправление вывода с команды создания резервной копии в SSH-клиент для передачи данных на сервер резервного копирования;
- **ssh** – SSH-клиент;
- me@backupserver – имя пользователя на сервере резервного копирования, имя самого сервера;
- "cat > /backup/fs_nutmeg/fs_search.sql" – команда, осуществляющая запись передаваемых через ssh-клиент данных в файл по указанному пути (путь указан для примера).

Перейдите на сервер резервного копирования, проверьте, что все необходимые файлы присутствуют.

4.2.7.2 Резервная копия объектного хранилища Swift

4.2.7.2.1 Установка rclone

Утилита **rclone** распространяется по лицензии MIT и предназначается для резервного копирования подсистемы объектного хранения данных.

Выполните установку утилиты на сервере резервного копирования по официальной инструкции: https://rclone.org/install/.

4.2.7.2.2 Настройка rclone

Настройка утилиты на работу с текущим экземпляром подсистемы объектного хранения данных может быть выполнена двумя способами:

- по официальной инструкции: https://rclone.org/swift/;
- созданием файла.

Порядок настройки вторым способом:

1. Создать файл в домашнем каталоге ~/.config/rclone/rclone.conf со следующим содержимым:

```
[fs]
type = swift
user = system:sync
key = ij5Zahzaew7jie8eiloh0nau2iete3oo
auth = http://192.168.1.15:8280/auth/v1.0
storage_url = http://192.168.1.15:8280/v1/AUTH_system
```

, где:

- **[fs]** – название соединения (потребуется далее);
- **type** = swift – константа для подключения к подсистеме объектного хранения данных;

- **user** – "имя аккаунта : имя пользователя";
- **key** – ключ учетной записи;
- **auth** – путь до прокси на подсистему объектного хранения данных. При этом нужно поменять IP-адрес и порт на необходимые;
- **storage_url** – путь до прокси на подсистему объектного хранения данных. При этом нужно поменять IP-адрес и порт на необходимые, изменить концовку адреса после AUTH_, которая должна соответствовать имени аккаунта.

2. Всю необходимую информацию можно найти на серверах роли **st** с помощью команды:

```
[root@st1 ~]# grep 'user_' /etc/swift/proxy-server.conf
user_system_sync = ij5Zahzaew7jie8eiloh0nau2iete3oo .admin
http://192.168.1.15:8280/v1/AUTH_system
```

, где:

- **user_system_sync** – составное имя аккаунта и имя пользователя с префиксом user_. Из данного примера получаем следующие параметры: имя аккаунта **system**, имя пользователя **sync** (получаем искомое: system:sync);
- ij5Zahzaew7jie8eiloh0nau2iete3oo – ключ;
- **.admin** – уровень прав доступа;
- http://192.168.1.15:8280/v1/AUTH_system – storage_url.

4.2.7.2.3 Выполнение резервного копирования объектного хранилища Swift

Для резервного копирования подсистемы объектного хранения данных:

1. Создать каталог для сохранения всех объектов подсистемы объектного хранения данных на сервере резервного копирования, перейти в каталог.

```
[root@backupserver ~]# mkdir /backup/fs_oldrelease/swift/
[root@backupserver  ~]# cd /backup/fs_oldrelease/swift/
```

2. Рекурсивно сохранить все объекты из Хранилища с помощью следующей команды:

```
[root@backupserver  ~]# for i in $(rclone lsf fs:); do rclone sync --progress fs:${i} ${i}; done
```

4.2.7.3 Резервная копия важных переменных для миграции

При переустановке Хранилища все конфигурационные файлы на серверах буду утеряны. Если не сделать резервные копии (а точнее: копии конкретных переменных) после переустановки и восстановления БД и подсистемы объектного хранения данных, то выполнить установку не получится из-за расхождения текущих солей и паролей с первоначальными.

Ниже приведен перечень переменных, которые необходимо сохранить, а также их расположение:

- токен хранилища **ETCD** – на серверах роли **database** в конфигурационном файле:

```
[root@db1 ~]# grep ETCD_INITIAL_CLUSTER_TOKEN /etc/etcd/etcd.conf | cut -d '=' -f 2 |
sed 's/"//g'
etcd-token
```

- пароли СУБД PostgreSQL – вместе с именами пользователей доступны для просмотра на серверах с ролью **backend** в конфигурационном файле

/etc/nct/nct-api.ini. Для Logos просмотр возможен через запущенное приложение (см. выше *«Поиск имен баз данных Файлового хранилища»*);

	Необязательно сохранять пароли от БД. Их можно создавать заново при установке Хранилища и прописывать во все конфигурационные файлы. Но при наличии возможности рекомендуется сохранить следующую информацию: пользователь **postgres**, пользователь мониторинга, пользователь репликации, пользователи к соответствующим базам данных.

- пароль хранилища структуры данных **Redis** (аналогично PostgreSQL: сохранять пароли необязательно, но можно) – получить пароль, если он задан, можно следующей командой:

```
[root@db1 ~]# grep -r requirepass /etc/redis/redis-6379.conf | cut -d ' ' -f 2
redis-password
```

- учетная запись к подсистеме объектного хранения данных – получить данные можно на серверах роли **storage** с помощью команды (отобразится: имя аккаунта, имя пользователя и ключ):

```
[root@st1 ~]# grep 'user_' /etc/swift/proxy-server.conf
user_system_sync = ij5Zahzaew7jie8eiloh0nau2iete3oo .admin
http://192.168.1.15:8280/v1/AUTH_system
```

- соли – всего значений солей 9. Все их необходимо сохранить и перенести в **group_vars**, соответствующей группы (см. п. 3.2.2.3, действия 7-10), и расставить согласно схеме соответствия (таблица 23);

Таблица 23 – Схема соответствия

Абсолютный путь до файла параметров на старой инсталляции на серверах с ролью fs_be	**Старое наименование параметра**	**Имя ключа в group_vars, значению которого должен быть равен параметр**

/etc/cginx.d/appapi.conf	salt1	fs.salts.salt1
/etc/cginx.d/appapi.conf	salt2	fs.salts.salt2
/etc/cginx.d/appapi.conf	salt3	fs.salts.salt3
/etc/cginx.d/appapi.conf	salt4	fs.salts.salt4
/etc/cginx.d/appapi.conf	salt5	fs.salts.salt5
/etc/cginx.d/appapi.conf	salt6	fs.salts.salt6
/etc/nct/nct-api.ini	salt_no_captcha	fs.salts.salt_no_captcha
/etc/nct/nct-api.ini	salt_no_auth	fs.salts.salt_no_auth
/etc/nct/nct-api.ini	salt	fs.salts.shared

	Если в указанных файлах не удалось найти всех искомых значений, значит используются значения по умолчанию, которые могут быть найдены в файле /usr/local/share/perl5/auto/share/dist/PSync-API/conf/nct-api.ini. Недостающие значения необходимо взять из этого файла. Если значения не указаны и там (может быть актуально для **salt1** и **salt5**), значит они равны «*».

- секретный ключ, вектор инициализации и соль для данных алгоритма AES-256-CBC, используемого для шифрования mail_session токена. Все их необходимо сохранить и перенести в **group_vars** соответствующей группы (см. п. 3.2.2.3 действия 7-10), и расставить согласно схеме соответствия (Таблица 24);

Таблица 24 – Схема соответствия

Абсолютный путь до файла параметров на старой инсталляции на серверах с ролью fs_fe	**Старое наименование параметра**	**Имя ключа в group_vars, значению которого должен быть равен параметр**
/etc/nginx/csrf.lua	key	fs.crypt.session.aes_key
/etc/nginx/csrf.lua	iv	fs.crypt.session.aes_iv
/etc/nginx/csrf.lua	salt	fs.crypt.session.salt

	В конфигурационном файле /etc/nginx/csrf.lua, искомые значения «обернуты» в функцию. Например: iv = fromhex("AEB2DB5F3F3F4B4E3289C94C507F89D6"); Важно понимать, что значение будет: "**AEB2DB5F3F3F4B4E3289C94C507F89D6**", а саму функцию нужно опустить.

- секретный ключ, вектор инициализации и соль для данных алгоритма AES-256-CBC, используемого для шифрования секретных данных тенантов.
  Все их необходимо сохранить и перенести в **group_vars**, соответствующей группы (см. п. 3.2.2.3, действия 7-10), и расставить согласно схеме соответствия (Таблица 25).

Таблица 25 – Схема соответствия

Абсолютный путь до файла параметров на старой инсталляции на серверах с ролью fs_be	**Старое наименование параметра**	**Имя ключа в group_vars, значению которого должен быть равен параметр**
/etc/cginx.d/appapi.conf	crypt_key	fs.crypt.main.aes_key
/etc/cginx.d/appapi.conf	crypt_iv	fs.crypt.main.aes_iv

4.3 Обновление Хранилища

4.3.1 Конфигурация без отказоустойчивости

4.3.1.1 Запуск обновления

Порядок запуска обновления описан в п. 4.3.2.3.

4.3.1.2 Проверка корректности обновления

Порядок проверки корректности обновления описан в п. 4.3.2.3.

4.3.1.3 Миграция данных

Порядок проверки корректности обновления описан в п. 4.3.2.3.

4.3.2 Кластерная отказоустойчивая конфигурация

4.3.2.1 Запуск обновления

Порядок запуска обновления описан в п. 4.3.2.3.

4.3.2.2 Проверка корректности обновления

Порядок проверки корректности обновления описан в п. 4.3.2.3.

4.3.2.3 Миграция данных

4.3.2.3.1 Восстановление из резервной копии

4.3.2.3.1.1 Восстановление важных переменных для миграции

Для восстановления Хранилища в первую очередь необходимо заполнить групповые переменные значениями из старой инсталляции, сохраненными по инструкции в п. 5.2.8. Все сохраненные значения необходимо перенести в **group_vars**, соответствующей группы (см. п. 3.2.2.3, действия 7-10).

	После восстановления переменных необходимо провести установку стенда «с нуля» по соответствующему разделу данной инструкции. После успешной установки можно приступать к последующим пунктам по восстановлению.

4.3.2.3.1.2 Восстановление баз данных СУБД PostgreSQL из резервной копии

Последовательность действий:

- узнать, какой сервер из кластера сейчас выполняет роль master СУБД PostgreSQL. Для этого потребуется зайти на любой сервер с ролью **fs_db** и выполнить команду:

```
[root@db1 ~]# docker exec -it stolon_fs0_keeper stolonctl status
```

Пример отображаемого вывода:

```
=== Active sentinels ===

ID      LEADER
924713f7    false
a30a301a    true
b7b2219c    false

=== Active proxies ===

ID

8444bcbf

b8801748

d6b2e8f3

d8d0e7a5

f25211d8

=== Keepers ===

UID         HEALTHY PG LISTENADDRESS    PG HEALTHY  PG WANTEDGENERATION  PG
CURRENTGENERATION

1fdb9789    true    fs-db-2:25432       true        3                    3

a409cb71    true    fs-db-3:25432       true        3                    3

b5bcbcb0    true    fs-db-1:25432       true        4                    4

=== Cluster Info ===

Master: b5bcbcb0

===== Keepers/DB tree =====

b5bcbcb0 (master)

--1fdb9789

--a409cb71
```

В секции "Cluster Info" отображается **UID** сервера с ролью **master** и выше в секции "Keepers" отображается соответствие **UID** и адреса сервера.

- скопировать все резервные копии баз данных СУБД PostgreSQL на сервер с ролью master;
- загрузить резервные копии баз данных в СУБД PostgreSQL (по команде на каждую БД):

```
[root@db1 ~]# docker exec -i stolon_fs0_keeper bash -c "PGPASSWORD='fs_search_password' psql -U db_search_user -h 172.17.0.1 fs_search" < fs_search.sql

[root@db1 ~]# docker exec -i stolon_fs0_keeper bash -c "PGPASSWORD='fs_vmail_password' psql -U db_vmail_user -h 172.17.0.1 fs_vmail" < fs_vmail.sql

[root@db1 ~]# docker exec -i stolon_fs0_keeper bash -c "PGPASSWORD='fs_password' psql -U db_fs_user -h 172.17.0.1 fs" < fs.sql

[root@db1 ~]# docker exec -i stolon_fs0_keeper bash -c "PGPASSWORD='lgs_password' psql -U db_lgs_user -h 172.17.0.1 lgs" < lgs.sql
```

После того, как были восстановлены базы данных, необходимо провести миграцию. Для этого на любом сервере роли **fs_be** нужно выполнить команду:

```
[root@db1 ~]# docker exec -it fs_webapi /usr/fsapi/bin/setup.pl --update --refresh
```

4.3.2.3.1.3 Восстановление подсистемы объектного хранения данных из резервной копии

Для восстановления подсистемы объектного хранения данных будет использоваться уже настроенная в п. 4.2.7.2 утилита **rclone**. Далее потребуется:

- на сервере резервного копирования необходимо перейти в директорию, где хранится резервная копия. Команда может выглядеть примерно так:

```
[root@backupserver  ~]# cd /backup/fs_oldrelease/swift/
```

- выполнить команду для восстановления данных в подсистеме объектного хранения данных:

```
[root@backupserver  ~]# for i in $(ls); do rclone sync --progress ${i} fs:${i}; done
```

4.3.2.3.1.4 Восстановление распределенного хранилища ETCD из резервной копии

После того, как были восстановлены СУБД и подсистема объектного хранения данных, необходимо выполнить восстановление распределенного хранилища **ETCD**.

Для этого:

- зайти на любой сервер с ролью **fs_be**;
- выполнить команду восстановления Хранилища:

```
[root@be1 ~]# docker exec -it fs_webapi bash -c "echo 'Y' | /usr/fsapi/bin/fix_script.pl --restore_etcd_data"
```

5 Техническая поддержка

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

Адрес электронной почты: support@service.myoffice.ru

Телефон: 8-800-222-1-888.

5.1 Системные сообщения

В случае использования неподдерживаемого браузера Система выводит соответствующее уведомление:

- на английском языке: «Your browser is not supported. Please use a different browser to run this app. Learn more about the browser support from restrictive documentation»;
- на русском языке: «Ваш браузер ограничивает работу приложения. Пожалуйста, воспользуйтесь другим браузером. Сведения о поддерживаемых браузерах приведены в сопроводительной документации».

5.2 Решение проблемы с установкой при использовании оборудования с низкой скоростью работы дисковой подсистемы

При проведении установки на таком оборудовании следует на этапе запуска скрипта установки fs_infra_*.run (см. п. 3.2.2) указать дополнительную опцию, лимитирующую количество потоков одновременной загрузки образов **Docker** в репозиторий. Значение параметра соответствует количеству одновременных потоков загрузки. По умолчанию значение равно "5".

Пример:

```
bash fs_infra_[RELEASE].run — --max-concurrent-uploads 2
```

Приложение 1 Подключение привязки тенанта к Active Directory

1.1 Организация доступа

Для работы системы аутентификации через Active Directory необходимо обеспечить доступ каждого узла сети be Хранилища ко всем контроллерам домена. Список разрешенных портов и соответствующие им сервисы:

- 389/tcp – ldap;
- 636/tcp – ldaps;
- 88/tcp, 88/udp – система аутентификации Kerberos;
- 750/udp – система аутентификации Kerberos version IV.

1.2 Конфигурация параметров group_vars подсистемы управления конфигурациями

Необходимо проверить и внести требующиеся изменения и дополнения в файл с именем установки. Такой файл был создан при выполнении пункта 3.2.2.3 настоящей Инструкции на шаге № 7.

1.2.1 Параметр "fs: kerberos:"

Проверить, что в настройках параметра "fs: kerberos:":

- флаг «enabled» имеет значение «True»;
- параметр «default_realm» описывает доменное имя или корневой домен;
- список «realms» содержит все необходимые домены и обслуживающие их контроллеры. Каждый элемент списка это параметр «name» со значением

имени домена и принадлежащий ему вложенный список соответствующих контроллеров «controllers».

Пример заполнения:

```
fs:
  kerberos:
    enabled: True
    default_realm: "domain.org"
    realms:
      - name: "test.local"
        controllers:
          - "ad.test.local"
      - name: "test2.local"
        controllers:
          - "ad1.test.local"
          - "ad2.test.local"
```

1.2.2 Параметр "fs: tenants:"

В Хранилище используется словарь, включающий описание всех требуемых тенантов. Требуется убедиться, что каждый теннант содержит следующие поля:

- «default_domain» описывает доменное имя или корневой домен, например: «domain.org»;
- «tariff_id» соответствует имени тарифа;
- «lang» соответствует поддерживаемому языку локализации (например, «ru-RU»);
- «recovery_email» содержит валидный адрес электронной почты администратора;
- секция «admin» описывает параметры учетной записи администратора, а именно:
  - «username»: имя пользователя;

- «firstname»: имя;
- «middlename»: отчество или второе имя;
- «lastname»: фамилия;
- «password»: пароль;

- секция «ad» описывает параметры подключения к **Active Directory** и зону поиска;
  - «base_dn»: базовое уникальное имя домена, например «DC=test,DC=local»;
  - – «default_realm»: область подключения по умолчанию, например «domain.org»;
  - «realm»:
- «url»: адрес сервера с указанием схемы подключения, например «ldap://ad.test.local».

Примеры заполнения двух тенантов: **default** и **second**:

```
fs:
  tenants:
    default:
      default_domain: "installation.example.org"
      tariff_id: "corp_1"
      lang: "ru-RU"
      recovery_email: "admin@company.org"
      admin:
        username: "fsadmin"
        firstname: "Ivan"
        middlename: "Ivanovich"
        lastname: "Ivanov"
        password: "password"
    second:
      default_domain: "domain.org"
```

```
  tariff_id: "corp_1"
  lang: "ru-RU"
  recovery_email: "admin@admin-net.org"
  admin:
    username: "fsadmin"
    firstname: "Ivan"
    middlename: "Ivanovich"
    lastname: "Ivanov"
    password: "password"
  ad:
    base_dn: "DC=test,DC=local"
    default_realm: "domain.org"
    realm: "DC=test,DC=local"
    url: "ldap://ad.test.local"
```

1.3 Применение конфигурации, запуск playbook подсистемы управления конфигурациями

Для того, чтобы применить конфигурацию, необходимо запустить **ansible-playbook** командой:

```
ansible-playbook playbooks/FS/main.yml --diff --tags fs_setup --limit fs_be
```

,где флаг "**diff**" включает отображение внесенных изменений во время выполнения, **tags** отвечает за выполнение сценария настройки, а **limit** ограничивает выполнение набора инструкций на узлах сети **be**.

МойОфис

Приложение 2 Поддержка виртуализации

2.1 Поддерживаемые системы виртуализации

Следующие системы виртуализации поддерживаются для обеспечения работы Хранилища:

- VMWare;
- KVM.

2.2 Рекомендации по размещению машин на гипервизорах

Для работы Хранилища в режиме отказоустойчивости на поддерживаемых системах виртуализации:

- запрещается размещение нескольких ролей **fs_db** или нескольких ролей **fs_st** или их комбинация на одном гипервизоре, с привязкой к одному физическому диску;
- запрещается размещение виртуальных машин ролей **fs_db** или **fs_st** в размере, превышающем необходимое количество машин, для установления кворума на одном гипервизоре;
- запрещается размещение всех виртуальных машин одной любой роли на одном гипервизоре.

2.3 Аппаратные средства в виртуальной среде

2.3.1 Минимальные требования

Минимальные требования для установки Хранилища на оборудовании без поддержки отказоустойчивости приведены в таблице 26.

Минимальные требования для установки Хранилища на отказоустойчивом оборудовании приведены в Таблице 27.

Таблица 26 – Минимальные требования (установка без поддержки отказоустойчивости)

Параметр	**Значение**
Процессор, vcpu	8
Оперативная память, Гб	24
Дисковая подсистема, Гб, тип	120, HDD + 40, SSD
Сетевой интерфейс, Мбит/сек	100
Количество серверов	1

Таблица 27 – Минимальные требования (отказоустойчивая установка)

Параметр	**fs_fe**	**fs_be**	**fs_db**	**fs_st**	**fs_infra**
Процессор, vcpu	1	2	2	2	2
Оперативная память, Гб	2	4	8	4	8
Дисковая подсистема, Гб, тип	20, HDD	20, HDD	40, SSD	40, HDD + 20 SSD	40, HDD
Сетевой интерфейс, Мбит/сек	1	1	1	1	1
Количество серверов	2	2	3	3	1

2.3.2 Рекомендованные требования

Рекомендованные требования для установки Хранилища на отказоустойчивом оборудовании приведены в таблице 28.

Таблица 28 – Рекомендованные требования

Параметр	fs_fe	fs_be	fs_db	fs_st	fs_infra
Процессор, vcpu	4	8	24	8	8
Оперативная память, Гб	8	16	128	16	32
Дисковая подсистема, Гб, тип	200, HDD	200, HDD	400, SSD	400, SSD + 16000, HDD	1000, SDD
Сетевой интерфейс, Мбит/сек	10	10	10	10	1
Количество серверов	2	4	3	3	1

Приложение 3 Примеры заполнения параметров групповых переменных для использования unbound

3.1 Установка без поддержки отказоустойчивости

Пример заполнения параметров групповых переменных при проведении установки без поддержки отказоустойчивости приведен в таблице 29.

Таблица 29 – Пример заполнения групповых переменных при установке без поддержки отказоустойчивости

Параметр	Описание	Тип	Значение по умолчанию
unbound:	Список подсетей, с которых можно обращаться к **unbound** для запросов разрешения имен		
access_control:	Параметры заполнения **access_control**		
- "172.17.0.0/16"	Обязательный параметр для разрешения обращений из docker-сети		
- "192.168.1.0/24"	Пример подсети, в которой находится сервер, с которого можно обращаться к **unbound**		
local_zone:	Параметры заполнения **local_zone**		
- type: "transparent"	Тип зоны		"transparent"
zone:	Домен по умолчанию для установки		"installation.example.net"
local_data:	A-записи для каждого сервера, в рамках которого проводится установка, и отдельно под **etcd** Пример заполнения записей при выполнении установки без поддержки отказоустойчивости		
- domain: "sa01.installation.example.net"			

	type: "A"			
	ip: "192.168.1.2"			
	- domain: "etcd.sa01.installation.example.net"			
	type: "A"			
	ip: "192.168.1.2"			

3.2 Кластерная установка

Пример заполнения параметров групповых переменных при проведении кластерной установки приведен в таблице 30.

Таблица 30 – Пример заполнения групповых переменных для отказоустойчивого оборудования

Параметр	Описание	Тип	Значение по умолчанию
unbound:	Список подсетей, с которых можно обращаться к **unbound** для запросов разрешения имен		
access_control:	Параметры заполнения **access_control**		
- "172.17.0.0/16"	Обязательный параметр для разрешения обращений из **docker**-сети		
- "192.168.1.0/24" - "192.168.2.0/24" - "192.168.3.0/24" - "192.168.4.0/24" - "192.168.5.0/24"	Пример подсетей, в которых находятся сервера, с которых можно обращаться к **unbound**		
local_zone:	Параметры заполнения **local_zone**		
- type: "transparent"	Тип		"transparent"
zone:	Домен по умолчанию для установки		"installation.example.net"

	local_data:	A-записи для каждого сервера, на котором проводится установка, и отдельно под **etcd** Пример заполнения записей при выполнении отказоустойчивой установки		
	- domain: "be01.installation.example.net"			
	type: "A"			
	ip: "192.168.1.2"			
	- domain: "be02.installation.example.net"			
	type: "A"			
	ip: "192.168.1.3"			
	- domain: "fe01.installation.example.net"			
	type: "A"			
	ip: "192.168.1.4"			
	- domain: "fe02.installation.example.net"			
	type: "A"			
	ip: "192.168.1.5"			
	- domain: "st01.installation.example.net"			
	type: "A"			
	ip: "192.168.1.6"			
	- domain: "st02.installation.example.net"			
	type: "A"			
	ip: "192.168.1.7"			
	- domain: "st03.installation.example.net"			
	type: "A"			
	ip: "192.168.1.8"			
	- domain: "db01.installation.example.net"			
	type: "A"			
	ip: "192.168.1.9"			

	- domain: "db02.installation.example.net"			
	type: "A"			
	ip: "192.168.1.10"			
	- domain: "db03.installation.example.net"			
	type: "A"			
	ip: "192.168.1.11"			
	- domain: "etcd.db01.installation.example.net"			
	type: "A"			
	ip: "192.168.1.9"			
	- domain: "etcd.db02.installation.example.net"			
	type: "A"			
	ip: "192.168.1.10"			
	- domain: "etcd.db03.installation.example.net"			
	type: "A"			
	ip: "192.168.1.11"			
	- domain: "infra01.installation.example.net"			
	type: "A"			

Если **unbound** не будет использоваться, в основных DNS-серверах необходимо дополнительно создать A-записи для:

- всех серверов, на которые будет производиться установка системы, по аналогии с примерами выше;
- **etcd** на сервера роли **fs-db**, по аналогии с примерами выше.

Внешние адреса необходимо создавать на основных DNS-серверах (авторитарных), отвечающих за разрешение имени основного домена установки, в соответствии с таблицей 31.

Таблица 31 – Требования к DNS

Запись	Тип	Значение	Комментарий
fsapi-<env>	A	<FS VIP web>	Адрес определяется **IPVS**, обслуживается

			через роль **FS LB**
admin-<env>	CNAME	fsapi-<env>.<public.domain>	
appapi-<env>	CNAME	fsapi-<env>.<public.domain>	Доступ не должен предоставляться внешним машинам
cardapi-<env>	CNAME	fsapi-<env>.<public.domain>	Актуально только для Nutmeg и выше
mailapi-<env>	CNAME	fsapi-<env>.<public.domain>	Актуально только для Nutmeg и выше
mtadmin-<env>	CNAME	fsapi-<env>.<public.domain>	
webbackupapi-<env>	CNAME	fsapi-<env>.<public.domain>	

МойОфис

Приложение 4 Расширенные параметры настройки

Далее приведены параметры, которые могут быть настроены дополнительно.

4.1 Параметры «ansible_user», «iptables»

Расширенные параметры настройки ОС приведены в таблице 32.

Таблица 32 – Параметры **ansible_user, iptables**

Параметр	Описание	Тип	Значение по умолчанию
ansible_user:	Указывается имя пользователя ssh-драйвера подсистемы управления конфигурациями	Str	admin
iptables:	Параметр настройки управление конфигурациями брандмауэра		
enabled:	Включение автоматической настройки службы управления межсетевым экраном. Параметр рекомендуется оставить включенным (True)	Bool	True
ssh_port:	Указывается порт **ssh**		
policy:	Блок параметров настройки правил для **iptables**		
input:	Указывается правило для цепочки **INPUT**		
output:	Указывается правило для цепочки **OUTPUT**		
forward:	Указывается правило для цепочки **FORWARD**		
custom_ruleset:	Указывается пользовательское правило		

Примеры корректно настроенных параметров:

```
ansible_user: "admin"
iptables:
  enabled: False
  ssh_port: 22
  policy:
```

```
  input: "DROP"
  output: "ACCEPT"
  forward: "DROP"
custom_ruleset: []
```

4.2 Параметр «chrony»

Расширенные параметры настройки серверного времени приведены в таблице 33.

Таблица 33 – Параметр **chrony**

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
chrony:	Параметр настройки серверного времени		
ntp:	Параметр настройки **ntp**		
servers:	Указывается список серверов для настройки серверного времени		

Примеры корректно настроенных параметров:

```
ntp:
 servers:
  - server1.ltd
  - server1.ltd
  - server1.ltd
```

4.3 Параметр «confd»

Расширенные параметры настройки шаблонов для настроенных сервисов приведены в таблице 34.

Таблица 34 – Параметр «confd»

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
confd:	Блок параметров настройки **confd**		
pattern:	Указывается регулярное выражение для определения toml/tmpl файлов, которые будут скопированы на хост		
conf_dir:	Указывается путь к настройкам **confd**		
backend:	Указывается база данных **ETCD**		

debug:	Включается (True) или отключается (False) параметр отладки
prefix:	Указывается префикс
watch:	Включается поддержка **watcher** (булево значение)
interval:	Указывается время отправления запроса к ETCD (в секундах)
resources:	Блок настройки **resources**
- service_name:	Указывается имя сервиса
src:	Указывается расширение сервиса
dest:	Указывается путь расположения сервиса
uid:	Указывается **uid**
gid:	Указывается **gid**
mode:	Указывается **mode**
key:	Указывается массив ключей
check_cmd:	Указывается команда проверки настроек
reload_cmd:	Указывается команда перезагрузки настроек

Примеры корректно настроенных параметров:

```
confd:
  pattern: "*"
  conf_dir: "/etc/confd"
  backend: "etcd"
  debug: False
  prefix: "/"
  watch: True
  interval: 60
  resources:
    - service_name: "nginx"
      src: "nginx.tmpl"
      dest: "/etc/nginx/sites-enabled/<appname>.conf"
      uid: 0
      gid: 0
      mode: "0644"
      key:
        - "/services/"
        - "/nginx"
      check_cmd: "/usr/sbin/nginx -t"
```

```
    reload_cmd: "/usr/sbin/service nginx reload"
```

4.4 Параметр «etcd»

Расширенные параметры настройки работы **etcd** приведены в таблице 35.

Таблица 35 – Параметр **etcd**

Параметр	Описание	Тип	Значение по умолчанию
etcd:	Блок параметров настройки **ETCD**		
auth:	Включается (True) или выключается (False) аутентификация		
password:	Указывается пароль к **ETCD**		
cluster:	Указывается список серверов		
use_tls:	При значении «True» будут созданы tls-сертификаты и ключи		
image:	Блок настроек **image**		
registry:	Указывается **registry**		
name:	Указывается имя		
tag:	Указывается тег		
settings:	Блок настроек **settings**		
heartbeat_interval:	Указывается **heartbeat_interval**		
election_timeout:	Указывается **election_timeout**		
snapshot_count:	Указывается **snapshot_count**		
quota_backend_bytes:	Указывается **quota_backend_bytes**		
max_snapshots:	Указывается **max_snapshots**		
max_wals:	Указывается **max_wals**		
enable_v2:	Включение (True) или выключение (False) параметра		
cert_auth:	Параметры настройки **cert_auth**		
peer:	Указывается **peer. Важно!** etcd.use_tls должно меть значение «True»		
client:	Указывается **client. Важно!** etcd.use_tls должно меть значение «True»		
token:	Указывается токен		

Примеры корректно настроенных параметров:

```
etcd:
```

```
auth: True
password: "super_secret"
cluster:
  - "host1"
  - "host2"
  - "host3"
use_tls: True
image:
  registry: "quay.io/coreos"
  name: "etcd"
  tag: "latest"
settings:
  heartbeat_interval: 100
  election_timeout: 1000
  snapshot_count: 10000
  quota_backend_bytes: 0
  max_snapshots: 5
  max_wals: 5
  enable_v2: True
  cert_auth:
    peer: True
    client: True
token: "test_env_cluster"
```

4.5 Параметр «fleet»

Расширенные параметры настройки подсистемы распределенного управления контейнерами приведены в таблице 36.

Таблица 36 – Параметр **fleet**

Параметр	Описание	Тип	Значение по умолчанию
fleet:	Блок параметров настройки **fleet**		
etcd:	Блок параметров настройки **ETCD**		
request_timeout:	Указывается **request_timeout**		
reconcile_interval:	Указывается **reconcile_interval**		

	agent_ttl:	Указывается **agent_ttl**		
	use_tls:	Включение (True) или выключение (False) **tls**		
	use_auth:	Включение (True) или выключение (False) аутентификации		

Примеры корректно настроенных параметров:

```
fleet:
  etcd:
    request_timeout: 1.0
    reconcile_interval: 2
    agent_ttl: "30s"
    use_tls: False
    use_auth: False
```

4.6 Параметр «kernel_ml»

Расширенные параметры настройки включения **elrepo**, удаления старой версии **kernel** и установки новой приведены в таблице 37.

Таблица 37 – Параметр **kernel_ml**

Параметр	Описание	Тип	Значение по умолчанию
kernel_ml:	Блок настройки параметров **kernel_ml**		
elrepo_repo_url:	Указывается ссылка на репозиторий **elrepo**		
elrepo_key_url:	Указывается ссылка на **elrepo_key**		

Примеры корректно настроенных параметров:

```
kernel_ml:
  elrepo_repo_url: "http://www.elrepo.org/elrepo-release-7.0-3.el7.elrepo.noarch.rpm"
  elrepo_key_url: "https://www.elrepo.org/RPM-GPG-KEY-elrepo.org"
```

4.7 Параметр «limits»

Расширенные параметры настройки ограничений для ОС приведены в таблице 38.

Таблица 38 – Параметр **limits**

Параметр	Описание	Тип	Значение по умолчанию
limits:	Блок параметров настройки **limits**		
default_options:	Параметры настройки опций по умолчанию		
- domain:	Указывается домен		
type:	Указывается тип		
item:	Указывается **item**		
value:	Указывается **value**		
filename:	Указывается имя файла		

Примеры корректно настроенных параметров:

```
limits:
  default_options:
    - domain: "*"
      type: "-"
      item: nofile
      value: 63536
      filename: "limits.conf"
```

4.8 Параметр «locale»

Расширенные параметры настройки языка и кодировки для каталогов приведены в таблице 39.

Таблица 39 – Параметр **locale**

Параметр	Описание	Тип	Значение по умолчанию
locale:	Блок параметров настройки **locale**		
system:	Параметры настройки системы		
locales:	Указывается **locales**		

Примеры корректно настроенных параметров:

```
system:
  locales: "en_US.utf8"
```

4.9 Параметр «nginx»

Расширенные параметры настройки **nginx** приведены в таблице 40.

Таблица 40 – Параметр **nginx**

Параметр	Описание	Тип	Значение по умолчанию
nginx:	Блок параметров настройки **nginx**		
image:	Параметры настройки **image**		
name:	Указывается имя сервера		
registry:	Указывается **registry**		
tag:	Указывается тег		
container:	Параметры настройки **container**		
ports:	Указываются порты		
conf:	Блок параметров настройки конфигурационных файлов **nginx**		
worker_processes:	Указывается число **worker_processes**		
worker_connections:	Указывается число **worker_connections**		
worker_rlimit_nofile:	Указывается число **worker_rlimit_nofile**		
use:	Указывается **use**		
sendfile:	Указывается **sendfile**		
tcp_nopush:	Указывается **tcp_nopush**		
tcp_nodelay:	Указывается **tcp_nodelay**		
server_tokens:	Указывается **server_tokens**		
gzip:	Указывается **gzip**		
keepalive_timeout:	Указывается число **keepalive_timeout**		
client_max_body_size:	Указывается **client_max_body_size**		
server_names_hash_bucket_size:	Задается размер **server_names_hash_bucket**		
logs:	Блок параметров настройки **logs**		
error:	Параметры настройки **error**		
path:	Указывается путь к **error.log**		

level:	Указывается **level**
access:	Параметры настройки **access**
path:	Указывается путь к **access.log**
format:	Указывается **format**
deployed_tls:	Блок параметров настройки **deployed_tls**
- cert_filename:	Указывается имя сертификата
key_filename:	Указывается имя ключа
ca_filename:	Указывается имя файла сертификата CA
ldap:	Блок параметров настройки **ldap**
enabled:	Включение (True) или выключение (False) параметра
image:	Параметр настройки **image**
name:	Указывается имя
tag:	Указывается тег
ports:	Указывается список портов
bind_dn:	Параметр настройки **bind_dn**
user:	Указывается пользователь
vhosts:	Блок параметров **vhosts**
server_name:	Указывается имя сервера
template:	Указывается шаблон
upstream:	Указывается адрес **upstream**
listen:	Указывается список listen серверов
ssl:	Блок параметров настройки **ssl**
redirect_to_https:	Включается (True) или выключается (False) переадресация
cert:	Указывается сертификат
key:	Указывается ключ

Примеры корректно настроенных параметров:

```
nginx:
  image:
    name: "nginx"
    registry: "library"
    tag: "alpine"
```

```
container:
  ports:
    - "80:80"
    - "443:443"
conf:
  worker_processes: "auto"
  worker_connections: 2048
  worker_rlimit_nofile: 10000
  use: "epoll"
  sendfile: "off"
  tcp_nopush: "on"
  tcp_nodelay: "on"
  server_tokens: "off"
  gzip: "on"
  keepalive_timeout: 30
  client_max_body_size: "1m"
  server_names_hash_bucket_size: 128
logs:
  error:
    path: "/var/log/nginx/error.log"
    level: "warn"
  access:
    path: "/var/log/nginx/access.log"
    format: "main"
deployed_tls:
  - cert_filename: "example.com-peer.pem"
    key_filename: "example.com-key.pem"
    ca_filename: "ca.example.com.pem"

ldap:
  enabled: False
  image:
    name: "registry.example.com/ldap-auth"
    tag: "latest"
  ports:
    - "8888:8888"

  bind_dn:
```

```
    user: "ucs-bot"
vhosts:
  "vhost1.example.com":
    server_name: "vhost1.example.com"
    template: "vhosts/proxy.conf.j2"
    upstream: "http://172.17.0.1:8080"
    listen:
      - "80 default_server"
      - "443 ssl http2 default_server"
    ssl:
      redirect_to_https: True
      cert: "bundle_example.com-peer.pem"
      key: "example.com-key.pem"
```

4.10 Параметр «postfix»

Расширенные параметры настройки **postfix** приведены в таблице 41.

Таблица 41 – Параметр **postfix**

Параметр	Описание	Тип	Значение по умолчанию
postfix:	Блок параметров настройки **postfix**		
image:	Параметры настройки **image**		
registry:	Указывается **registry**		
tag:	Указывается тег		
name:	Указывается имя		
uid:	Указывается **uid**		
gid:	Указывается **gid**		
relay:	Параметры настройки роли **relay**		
container:	Параметры настройки **container**		
bind_ip:	Указывается **bind_ip**		
bind_port:	Указывается **bind_port**		25
log_driver:	Указывается **log_driver**		
log_options:	Параметры настройки **log_options**		
syslog-address:	Указывается **syslog-address**		

tag:	Указывается тег
use_tls:	Включается (True) или выключается (False) использование **tls**
tls_mode:	Указывается режим
tls:	Параметры настройки **tls**
cert_filename:	Указывается имя сертификата домена
key_filename:	Указывается имя файла ключа для сертификата домена
ca_filename:	Указывается имя файла сертификата CA
relayhost:	Параметры настройки **relayhost**
name:	Указывается имя
port:	Указывается порт 25
milter:	Параметры настройки **milter**
enabled:	Включение (True) или выключение (False) параметра
endpoints:	Указываются **endpoints**
parameters:	Параметры настройки **parameters**
custom:	Указывается **custom**
default:	Параметры настройки по умолчанию
mydomain:	Указывается адрес домена
relay_domains:	Указывается список доменов
mynetworks:	Указываются **networks**
restrictions:	Параметры настройки **restrictions**
client:	Указывается **client**
helo:	Указывается **helo**
sender:	Указывается **sender**
recipient:	Указывается **recipient**
limits:	Параметры настройки **limits**
anvil_rate_time_unit:	Указывается **anvil_rate_time_unit**
anvil_status_update_time:	Указывается **anvil_status_update_time**
bounce_size_limit:	Указывается **bounce_size_limit**
default_process_limit:	Указывается **default_process_limit**

	lmtp_destination_concurrency_limit:	Указывается **lmtp_destination_concurrency_limit**
	max_use:	Указывается **max_use**
	message_size_limit:	Указывается **message_size_limit**
	smtpd_client_message_rate_limit:	Указывается **smtpd_client_message_rate_limit**
	smtpd_client_connection_count_limit:	Указывается **smtpd_client_connection_count_limit**
mx:		Параметры настройки роли **mx**
use_tls:		Включение (True) или выключение (False) параметра
use_woof_ldap:		Включение (True) или выключение (False) параметра
milter:		Параметры настройки **milter**
enabled:		Включение (True) или выключение (False) параметра | | False
parameters:		Параметры настройки **parameters**
custom:		Указывается **custom**
default:		Параметры настройки **default**
relay_domains:		Указывается список доменов
mynetworks:		Указываются **networks**
restrictions:		Параметры настройки **restrictions**
	client:	Указывается **client**
	helo:	Указывается **helo**
	sender:	Указывается **sender**
	recipient:	Указывается **recipient**
limits:		Параметры настройки **limits**
	anvil_rate_time_unit:	Указывается **anvil_rate_time_unit**
	anvil_status_update_time:	Указывается **anvil_status_update_time**
	bounce_size_limit:	Указывается **bounce_size_limit**
	default_process_limit:	Указывается **default_process_limit**
	lmtp_destination_concurrency_limit:	Указывается **lmtp_destination_concurrency_limit**
	max_use:	Указывается **max_use**
	message_size_limit:	Указывается **message_size_limit**

	smtpd_client_message_rate_limit:	Указывается **smtpd_client_message_rate_limit**
	smtpd_client_connection_count_limit:	Указывается **smtpd_client_connection_count_limit**

Примеры корректно настроенных параметров:

```
postfix:
 image:
   registry: "{{ docker.registry.endpoint }}"
   tag: "latest"
   name: "nct_postfix"
   # uid of postfix user in container
   uid: 89
   # gid of postfix user in container
   gid: 89
 relay:
   container:
     bind_ip: "172.17.0.1"
     bind_port: 10025
logging-drivers
     log_driver: "syslog"
     log_options:
       syslog-address: "udp://{{ inventory_hostname }}:514"
       tag: "postfix_relay"
   use_tls: True
   tls_mode: "starttls"
   tls:
     cert_filename: "certificate-peer.pem"
     key_filename: "certificate-key.pem"
     ca_filename: "ca-certificate.pem"
   relayhost:
     name: "10.7.97.13"
     port: 26
   milter:
     enabled: True
     endpoints:
       - "inet:milter1.example.local"
   parameters:
```

```
    custom: []
    default:
      mydomain: "relay-server.nct.local"
      relay_domains: []
      mynetworks: []
      restrictions:
        client: []
        helo: []
        sender: []
        recipient:
          - "reject_unauth_destination"
      limits:
        anvil_rate_time_unit: "10s"
        anvil_status_update_time: "600s"
        bounce_size_limit: 10000
        default_process_limit: 100
        lmtp_destination_concurrency_limit: 50
        max_use: 600
        message_size_limit: "{{ services.mail_limit_size_bytes }}"
        smtpd_client_message_rate_limit: 100
        smtpd_client_connection_count_limit: 50
mx:
  use_tls: True
  use_woof_ldap: False
  milter:
    enabled: False
  parameters:
    custom: []
    default:
      relay_domains: []
      mynetworks: []
      restrictions:
        client: []
        # Default for mx:
        helo:
          - "reject_non_fqdn_hostname"
          - "reject_unknown_hostname"
          - "reject_invalid_hostname"
```

```
      - "reject_unknown_client"
    # Default for mx:
    sender:
      - "reject_non_fqdn_sender"
      - "reject_unknown_sender_domain"
    # Default for mx:
    recipient:
      - "reject_unauth_pipelining"
      - "reject_unknown_client"
      - "reject_unknown_recipient_domain"
      - "reject_non_fqdn_recipient"
      - "reject_unauth_destination"
  limits:
    anvil_rate_time_unit: "10s"
    anvil_status_update_time: "600s"
    bounce_size_limit: 10000
    default_process_limit: 100
    lmtp_destination_concurrency_limit: 50
    max_use: 600
    message_size_limit: "{{ services.mail_limit_size_bytes }}"
    smtpd_client_message_rate_limit: 100
    smtpd_client_connection_count_limit: 50
```

4.11 Параметр «redis»

Расширенные параметры настройки **Redis** приведены в таблице 42.

Таблица 42 – Параметр **Redis**

Параметр	Описание	Тип	Значение по умолчанию
redis:	Блок настроек хранилища **Redis**		
cluster:	Параметр настройки кластера		
name:	Указывается наименование кластера		
master:	Параметр настройки кластера **master**		
password:	Указывается пароль		
node_number:	Указывается номер кластера **master** хранилища **Redis**, например, "01"		
hosting_server:	Указывается **Hostname** сервера, на котором будет размещено хранилище **Redis**. Параметр должен соответствовать значению		

	hostname в файле **inventory**
slaves:	Указывается список узлов сети **slave**[10]
- node_number:	Указывается номер узла сети кластера, например, "01"
hosting_server:	Указывается **Hostname** сервера, на котором будет размещен кластер
client_port:	Указывается порт **client**
sentinel:	Параметры настройки **sentinel**
quorum_number:	Указывается **quorum_number**
down_after_ms:	Указывается **down_after_ms**
failover_timeout:	Указывается **failover_timeout**
parallel_syncs:	Указывается **parallel_syncs**
cluster:	Параметры настройки кластера
- node_number:	Указывается номер узла сети кластера, например, "01"
hosting_server:	Указывается **Hostname** сервера, на котором будет размещен кластер
client_port:	Указывается порт **client**
tunnel:	Параметры настройки **tunnel**
nodes:	Параметры настройки **nodes**
- node_number:	Указывается номер узла сети кластера, например, "01"
hosting_server:	Указывается **Hostname** сервера, на котором будет размещен кластер
databases:	Параметры настройки **databases**
name:	Указывается имя базы данных
local_port:	Указывается порт

Примеры корректно настроенных параметров:

```
redis:
  cluster:
     password: "test_password"
     node_number: "01"
     hosting_server: "srv1.example.com"
    slaves:
     - node_number: "02"
       hosting_server: "srv2.example.com"
       client_port: 6380
```

[10] Только для установки в отказоустойчивом режиме.

МойОфис

```
    - node_number: "03"
      hosting_server: "srv3.example.com"
      client_port: 6381
sentinel:
  quorum_number: 2
  down_after_ms: 3000
  failover_timeout: 10000
  parallel_syncs: 1
  cluster:
    - node_number: "01"
      hosting_server: "srv1.example.com"
    - node_number: "02"
      hosting_server: "srv2.example.com"
      client_port: 26380
    - node_number: "03"
      hosting_server: "srv3.example.com"
      client_port: 26381
  tunnel:
    nodes:
      - node_number: "01"
        hosting_server: "be-srv1.example.com"
        databases:
          - name: "nctmaster"
            local_port: "6369"
          - name: "nctextend"
            local_port: "6370"
      - node_number: "02"
        hosting_server: "be-srv2.example.com"
        databases:
          - name: "nctextra"
            local_port: "6371"
```

4.12 Параметр «resolv»

Параметры **resolv** приведены в таблице 43.

Таблица 43 – Параметр **resolv**

Параметр	Описание	Тип	Значение по умолчанию
resolv:	Блок параметров настройки **resolv**		
search:	Указывается **search**		
domain:	Указывается IP-адрес		
nameserver:	Указывается **nameserver**		
sortlist:	Указывается **sortlist**		
options:	Выбирается одна из доступных опций: • "debug" – устанавливает RES_DEBUG в _res.options; • "ndots:n" – устанавливает предел для количества точек, которые должны появиться в имени для res_query до того, как будет выполнен запрос; • "rotate" – устанавливает RES_ROTATE в _res.options; • "attempts:n" – устанавливает временной интервал, в ходе которого преобразователь (resolver) будет отправлять запросы к серверам		

Примеры корректно настроенных параметров:

```
resolv:
  search: []
  domain: "srv1.example.com"
  nameserver: []
  sortlist: []
  options:
    - "debug"
    - "ndots:1"
    - "rotate"
    - "attempts:2"
```

4.13 Параметр «rsyslog»

Расширенные параметры настройки **rsyslog** для работы с **Docker** приведены в таблице 44.

Таблица 44 – Параметр **rsyslog**

Параметр	Описание	Тип	Значение по умолчанию
rsyslog:	Блок параметров настройки **rsyslog**		
docker:	Параметры настройки **docker**		
enabled:	Включение (True) или выключение (False) параметра		
docker_socket:	Указывается **docker_socket**		
container_socket:	Указывается **container_socket**		
forward:	Параметры настройки **forward**		
enabled:	Включение (True) или выключение (False) параметра		
template:	Указывается **template**		
targets:	Указываются**targets**		

Примеры корректно настроенных параметров:

```
rsyslog:
  docker:
    enabled: False
    docker_socket: "/var/run/rsyslog/docker.sock"
    container_socket: "/var/run/container_rsyslog/rsyslog.sock"

  forward:
    enabled: False
    template: "json"
    targets:
      - "host.domain.ltd:port"
```

4.14 Параметр «SElinux»

Параметры **SElinux** приведены в таблице 45.

Таблица 45 – Параметр **SElinux**

Параметр	Описание	Тип	Значение по умолчанию
SElinux:	Параметр настройки **SElinux**		
state:	Выбрать одно из состояний: • enforcing; • permissive; • disabled		enforcing

Примеры корректно настроенных параметров:

```
selinux:
  state: enforcing
```

4.15 Параметр «ssh_keys»

Расширенные параметры настройки ssh-ключей приведены в таблице 46.

Таблица 46 – Параметр **ssh_keys**

Параметр	Описание	Тип	Значение по умолчанию
ssh_keys:	Блок параметров настройки **ssh_keys**		
system_users:	Параметр настройки **system_users**		
- name:	Указывается имя		
authorized:	Параметр настройки **authorized**		
- name:	Указывается имя		
public_key:	Указывается **public_key**		

Примеры корректно настроенных параметров:

```
ssh_keys:
  system_users:
    - name: "centos"
```

```
    authorized:
      - name: "username_1"
        public_key: "ssh-rsa AAAAB3NzaC...EWawg3 username_1@localhost.localdomain"
      - name: "username_2"
        public_key: "ssh-rsa AAAYGGHsNzaC...EWawg3 username_2@localhost.localdomain"
  - name: "root"
    authorized:
      - name: "root_username_1"
        public_key: "ssh-rsa AAAAB3NzaC...EWawg3 root_username_1@localhost.localdomain"
```

4.16 Параметр «sshd»

Расширенные параметры настройки конфигурационного файла **sshd** приведены в таблице 47.

Таблица 47 – Параметр **sshd**

Параметр	Описание	Тип	Значение по умолчанию
sshd:	Блок параметра настройки **sshd**		
protocol:	Указывается номер протокола		2
accept_env:	Указывается **accept_env**		
permit_root_login:	Включается (yes) или выключается (no) **permit_root_login**		yes
password_authentication:	Включается (yes) или выключается (no) **password_authentication**		yes
use_dns:	Включается (yes) или выключается (no) **use_dns**		yes
x11_forwarding:	Включается (yes) или выключается (no) **x11_forwarding**		no

	allow_groups:	Указывается список **allow_groups**
	allow_users:	Указывается список **allow_users**

Примеры корректно настроенных параметров:

```
sshd:
  protocol: 2
  accept_env: "LC_*"
  permit_root_login: "no"
  password_authentication: "yes"
  use_dns: "no"
  x11_forwarding: "no"
  allow_groups: []
  allow_users: []
```

4.17 Параметр «stolon»

Расширенные параметры настройки **stolon** приведен в таблице 48.

Таблица 48 – Параметр **stolon**

Параметр	Описание	Тип	Значение по умолчанию
stolon:	Блок параметров настройки **stolon**		
cluster_name:	Указывается наименование кластера		
etcd_endpoints:	Указываются **endpoints ETCD**		
postgres_password:	Указывается пароль Postgres		
replication:	Параметры настройки копий		
port:	Указывается порт		
username:	Указывается имя пользователя		
password:	Указывается пароль		
proxy:	Параметры настройки прокси		
port:	Указывается порт		
image:	Параметры настройки **image**		
registry:	Указывается **registry**		
tag:	Указывается тег		
postgres_uid:	Указывается **uid** Postgres		

postgres_gid:	Указывается **gid** Postgres
users:	Параметры настройки пользователей
name:	Указывается имя пользователя
password:	Указывается пароль пользователя
databases:	Параметры настройки базы данных
name:	Указывается имя базы данных
owner:	Указывается имя пользователя, который является владельцем базы данных

Примеры корректно настроенных параметров:

```
stolon:
  cluster_name: "postgres0"
  etcd_endpoints:
    - "http://etcd.example.com"
  postgres_password: "CorrectHorseBatteryStaple"
  replication:
    port: 25432
    username: "repluser"
    password: "replpassword"
  proxy:
    port: 5432
  image:
    registry: "registry.example.com"
    tag: "latest"
    postgres_uid: 999
    postgres_gid: 999
  users:
    - name: "some_user"
      password: "changeme"
  databases:
    - name: "some_database"
      owner: "some_user"
```

4.18 Параметр «swift»

Расширенные параметры настройки **Swift** приведены в таблице 49.

Таблица 49 – Параметр **Swift**

Параметр	Описание	Тип	Значение по умолчанию
swift:	Блок параметров настройки **Swift**		
uid:	Указывается **uid**		
gid:	Указывается **gid**		
image:	Параметры настройки **image**		
registry:	Указывается **registry**		
name:	Указывается имя		
tag:	Указывается тег		
ring:	Параметры настройки **ring**		
part_power:	Указывается **part_power**		
replicas:	Указывается количество копий		
min_part_hours:	Указывается **min_part_hours**		
node_dir:	Указывается **node_dir**		
cluster:	Параметры настройки кластера		
name:	Указывается имя		
nodes:	Параметры настройки узлов кластера		
- node_number:	Указывается номер узла кластера. Например, "01"		
hosting_server:	Указывается **Hostname** сервера, на котором будет размещен узел. Параметр должен соответствовать **hostname** в файле **inventory**		
drives:	Указывается список дисков для хранения данных. Диск должен быть примонтирован в раздел /srv/node/<drive_name>		
ports:	Параметры настройки портов		
replication:	Указывается копия		
ports:	Параметры настройки портов		
account_server:	Указывается **account_server**		
container_server:	Указывается **container_server**		
object_server:	Указывается **object_server**		

	replication:	Указывается **replication**
	proxy_server:	Указывается **proxy_server**
	memcached:	Указывается **memcached**
	users:	Параметры настройки пользователей
	- account:	Указывается **account**
	user:	Указывается пользователь
	key:	Указывается **key**
	groups:	Указывается группы

Примеры корректно настроенных параметров:

```
swift:
  uid: 160
  gid: 160
  image:
    registry: "registry.example.com"
    name: "swift"
    tag: "latest"
  ring:
    part_power: 10
    replicas: 3
    min_part_hours: 1
  node_dir: "/srv/node"
  cluster:
    name: "my_awesome_cluster"
    nodes:
      - node_number: 01
        hosting_server: storage01.example.com
          - data0
          - sdc
        ports:
          replication: 874
  ports:
    account_server: 6002
    container_server: 6001
    object_server: 6000
    replication: 873
```

```
  proxy_server: 8280
  memcached: 11211
users:
  - account: "admin"
    user: "admin"
    key: "changeme"
    groups:
      - ".admin"
      - ".reseller_admin"
```

4.19 Параметр «sysctl»

Расширенные параметры настройки **sysctl** приведены в таблице 50.

Таблица 50 – Параметр **sysctl**

Параметр	Описание	Тип	Значение по умолчанию
sysctl:	Блок параметров настройки **sysctl**		
default_sysctl_options:	Параметры настройки **default_sysctl_options**		
value:	Указывается **value**		

Примеры корректно настроенных параметров:

```
sysctl:
  default_sysctl_options:
    fs.file-max:
      value: 9897299
    net.core.somaxconn:
      value: 65535
    vm.max_map_count:
       value: 1048575
```

4.20 Параметр «timesyncd»

Данные параметры приведены в таблице 51.

Таблица 51 – Параметр **timesyncd**

Параметр	Описание	Тип	Значение по умолчанию

timesyncd:	Блок параметров настройки **timesyncd**
- name:	Указывается имя
systemd:	Параметры настройки **systemd**
name:	Указывается имя
state:	Указывается состояние
enabled:	Включение (True) или выключение (False) параметра
masked:	Включение (True) или выключение (False) маски

Примеры корректно настроенных параметров:

```
- name: "ensure ptp is stopped and masked"
  systemd:
    name: "timemaster"
    state: stopped
    enabled: False
    masked: True
```

4.21 Параметр «timezone»

Расширенные параметры настройки часового пояса для сервера приведены в таблице 52.

Таблица 52 – Параметр **timezone**

Параметр	Описание	Тип	Значение по умолчанию
timezone:	Блоку параметров настройки **timezone**		
system:	Настройка системы		
timezone:	Указывается **timezone**		

Примеры корректно настроенных параметров:

```
system:
  timezone: "Europe/Moscow"
```

4.22 Параметр «unbound»

Расширенные параметры настройки **unbound** приведены в таблице 53. Параметр работает только на ОС Astra Linux.

Таблица 53 – Параметр **unbound**

Параметр	Описание	Тип	Значение по умолчанию
unbound:	Блок параметров настройки **unbound**		
listen_interfaces:	Параметр настройки **listen_interfaces**		
local_ipv4:	Указывается **local_ipv4**		
default:	Указывается **default**		
docker:	Параметр настройки		
used:	Указывается **used**		
interface:	Указывается **interface**		
extra:	Указывается **extra**		
access_control:	Указывается **access_control**		
forward_dns:	Указывается список **forward_dns**		
local_zone:	Параметр настройки **local_zone**		
- type:	Указывается тип		
zone:	Указывается зона		
local_data:	Параметр настройки **local_data**		
- domain:	Указывается домен		
type:	Указывается тип		
ip:	Указывается ip-адрес		

Примеры корректно настроенных параметров:

```
unbound:
  listen_interfaces:
    local_ipv4: True
    default: True
    docker:
      used: True
      interface: "172.17.0.1"
    extra: []
  access_control:
    - "192.168.103.0/24"
  forward_dns:
    - "8.8.8.8"
    - "8.8.4.4"
  local_zone:
    - type: "nodefault"
      zone: "10.in-addr.arpa"
    - type: "transparent"
      zone: "10.in-addr.arpa"
      local_data:
        - domain: "example.com"
          type: "A"
          ip: "1.2.3.4"
    - type: "redirect"
      zone: "10.in-addr.arpa"
      local_data:
        - domain: "example.com"
          type: "A"
          ip: "1.2.3.4"
```

4.23 Параметр «yum»

Параметры настройки **yum** приведены в таблице 54.

Таблица 54 – Параметр **yum**

Параметр	Описание	Тип	Значение по умолчанию
yum:	Блок параметров настройки **yum**		
configure:	Рекомендуемое значение: False При значении True менеджер обратиться к шаблону настроек		
cron:	Указывается True или False		
conf:	Параметр настройки **conf**		
installonly_limit:	Указывается **installonly_limit**		
set_timeout:	Указывается True или False		
timeout:	Указывается **timeout**		
refresh_cache:	Рекомендуемое значение: False При значении True команда 'yum makecache' будет запущена		
update:	Рекомендуемое значение: False При значении True команда 'yum update *' будет запущена		

Примеры корректно настроенных параметров:

```
yum:
  configure: False
  cron: False
  conf:
    installonly_limit: 3
    set_timeout: True
    timeout: 5
  refresh_cache: False
  update: False
```