Файловое хранилище

Инструкция по развертыванию

2019.02 Oregano

Содержание

Аннотация

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.

Что изменилось?

Доработана функциональность:

- обновлен пакет установки, добавлены компоненты Хранилище файлов, Администрирование на Ansible;
- устранен код, обращающийся к неиспользуемой почтовой подсистеме, перенесены все необходимые функции для корректной работы связанных компонент;
- вкладка **Пользователи** компонента Администрирование перенесена на новую архитектуру;
- вкладка **Компания** компонента Администрирование перенесена на новую архитектуру;
- произведена локализация Web Admin (добавлен испанский и французский языки);
- реализована возможность задания квоты на количество пользователей и объем дискового пространства при создании тенанта;
- реализована возможность задания настроек SMTP-сервера для отправки системных сообщений и работы формы обратной связи;
- реализована возможность просмотра для пользователя с соответствующим набором прав (re-share) списка пользователей, которым доступен объект, а также управление этим списком в приложении Документы;
- реализована возможность удалять отмеченных на удаление пользователей как по расписанию, так и по запросу, без возможности восстановления;
- выделен отдельный код ошибки в случае попытки создания пользователя при превышенной квоте;
- добавлены ограничения на ввод символов в поля **ФИО**;
- реализовано предоставление по запросу списка всех доменов;
- улучшен код для оптимизации работы приложения.

Исправлены ошибки:

- невозможность смены порта в настройках SMTP-сервера, который используется для отправки системных сообщений;
- пользователям с соответствующими правами убран доступ к файлам, владельцы которого находятся в заблокированном тенанте;
- удаленные файлы больше не выдаются в результатах поиска;
- поле **Кому** в системных письмах отображает имя пользователя, а не email адрес;

- при удалении администратором тенанта пользователя с ролью **Администратор**, если такой пользователь авторизован в системе, производится автоматический выход из системы;
- при изменении прав пользователя с ролью **Администратор** другим администратором, а также при блокировке такого пользователя, если такой пользователь авторизован и активен в системе, ему недоступны для просмотра данные форм (список пользователей, список групп);
- домен для основного email пользователя отображается всегда;
- пользователи, добавленные во вновь созданную группу, отображаются в списке результатов поиска;
- при превышении квоты по количеству пользователей нельзя добавить нового AD пользователя;
- при удалении дополнительного домена, используемого в алиасе пользователя, выводится предупреждение с предложением выбора домена для замены;
- срок действия пароля не истекает при привязке и дальнейшей отвязки учетной записи ЕСИА;
- ошибка сервера не возникает при вводе нуля в диалоговое окно предоставления доступа к документу.

1 Введение

Настоящая инструкция (далее – Инструкция) описывает последовательность действий для установки Файлового хранилища (далее – Хранилище).

2 Системные требования

2.1 Требования к квалификации персонала

Администратор, выполняющий действия по настоящей Инструкции, должен обладать следующими навыками и знаниями:

- 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.

2.2 Аппаратные средства

Ниже представлены минимальные и рекомендованные требования.

2.2.1 Минимальные требования

Минимальные требования для установки Хранилища на оборудовании без поддержки отказоустойчивости приведены в таблице 1.

Данный режим работы предоставляется в целях демонстрации функциональности системы. Данный режим не поддерживается, не рекомендуется его использовать.

В данном режиме все роли устанавливаются на один физический сервер.

Минимальные требования для установки Хранилища на отказоустойчивом оборудовании приведены в таблице 2.

Таблица 1 — 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 2 — 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	20, HDD	20, HDD	40, SSD	40, HDD +	40, HDD

подсистема, Гб, тип				20 SSD	
Сетевой интерфейс, Мбит/сек	1	1	1	1	1
Количество серверов	2	2	3	3	1

2.2.2 Рекомендованные требования

Рекомендованные требования для установки Хранилища на отказоустойчивом оборудовании приведены в таблице 3.

Таблица 3 — 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

2.3 Программное обеспечение

Требования к программному обеспечению для места оператора и оборудования, на котором производится установка, приведены в таблицах 4-5.

Таблица 4 — Требования к программному обеспечению для места оператора

Требование	Описание
Поддерживаемые браузеры	• Chrome – не ниже версии 74; • Microsoft Edge – не ниже версии18; • 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+
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
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)

В hosts.yml (в файле inventory ansible) вносятся только доменные имена. Часть логики установщика использует их для формирования доменных имен и адресов сервисов. Запрещено вносить IP-адреса!

В системе не должно быть других конфигурационных файлов ansible. 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

Таблица 5 — 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"

Отсутствует поддержка систем виртуализации!

2.3.1 Рекомендации по разбивке дисков

Рекомендации по разбивке дисков приведены в таблице 6.

Таблица 6 — Рекомендации по разбивке дисков

Сервера роли	Рекомендация
fs_fe	На машинах хранятся только статические файлы веб-интерфейса. Рекомендуется минимальный размер диска, необходимый для штатной работы ОС (не менее 20 Гб)
fs_be	На машинах хранятся только образы всех запущенных контейнеров. Рекомендуется минимальный размер диска, необходимый для штатной работы ОС (не менее 20 Гб)
fs_db	На машинах хранятся базы данных **redis** и **postgres** в каталоге /srv/docker. Журналы всех запущенных контейнеров попадают в файл /var/log/containers с параметром rotate=2. Под каталог /srv/ требуется выделить отдельный раздел (не менее 10 Гб, 80% от места на диске. Рекомендуется использовать SSD)
fs_st	На машинах хранятся файлы объектного хранилища **swift** (занимает больше всего места в Хранилище). Данные **swift** по умолчанию записываются в каталог /srv/node. При использовании выделенных дисков для хранения файлов (рекомендуется), их необходимо смонтировать в каталог /srv/node/sd[a-z]. Рекомендуется выделить под каталог /srv/ отдельный раздел с минимальным объемом 20 Гб. Увеличивать раздел по необходимости с учетом ожидания роста объектного хранилища
fs_infra	На машинах хранятся **docker registry** и пакетные репозитории. В будущих релизах здесь будет реализованы: коллектор журналов, сервер мониторинга и оповещения. Все данные будут храниться в /srv/docker. Журналы всех запущенных контейнеров попадают в файл /var/log/containers с параметром rotate=2. Создать отдельный раздел под /var/log. Рекомендуется выделить под каталог /srv/ отдельный раздел с минимальным объемом 10 Гб. Увеличивать раздел по необходимости с учетом ожидания роста нагрузки на систему и соответственного увеличения объема журналов работы системы

2.3.2 Требования к DNS

2.3.2.1 Внутренние адреса

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

> **unbound** используется для запросов внутри системы и подключается только для контейнеров, через соответствующие параметры групповых переменных.
>
> **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 заполнение файла с групповыми переменными, выставляются параметры **docker** на основные DNS-сервера, **unbound** будет установлен и настроен, но не будет принимать участия в работе системы.

Примеры заполненных групповых переменных для использования **unbound** в рамках установки без поддержки отказоустойчивости приведены в таблице 7.

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

Таблица 7 — Пример заполнения групповых переменных (установка без поддержки отказоустойчивости)

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
unbound:	Список подсетей, с которых можно обращаться к **unbound** для запросов разрешения имен		
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:				
			- 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"			

Примеры заполненных групповых переменных с целью использования **unbound** для отказоустойчивой установке приведены в таблице 8.

Таблица 8 — Пример заполнения групповых переменных (отказоустойчивая установка)

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
unbound:	Список подсетей, с которых можно обращаться к **unbound** для запросов разрешения имен		
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:			
- 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: "be01.installation.example.net"			
				type: "A"			
				ip: "192.168.1.2"			
				- domain: "be02.installation.example.net"			
				type: "A"			
				ip: "192.168.1.3"			
				- domain: "be01.installation.example.net"			
				type: "A"			
				ip: "192.168.1.2"			
				- domain: "fe01.installation.example.net"			
				type: "A"			
				ip: "192.168.2.2"			
				- domain: "fe02.installation.example.net"			
				type: "A"			
				ip: "192.168.2.3"			
				- domain: "st01.installation.example.net"			

				type: "A"			
				ip: "192.168.3.2"			
				- domain: "st02.installation.example.net"			
				type: "A"			
				ip: "192.168.3.3"			
				- domain: "st03.installation.example.net"			
				type: "A"			
				ip: "192.168.3.4"			
				- domain: "db01.installation.example.net"			
				type: "A"			
				ip: "192.168.4.2"			
				- domain: "db02.installation.example.net"			
				type: "A"			
				ip: "192.168.4.3"			
				- domain: "db03.installation.example.net"			
				type: "A"			
				ip: "192.168.4.4"			

				- domain: "etcd.db01.installation.example.net"			
				type: "A"			
				ip: "192.168.4.4"			
				domain: "infra01.installation.example.net"			
				type: "A"			
				ip: "192.168.5.2"			

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.

2.3.2.2 Внешние адреса

Внешние адреса необходимо создавать на основных DNS-серверах (авторитарных), отвечающих за разрешение имени основного домена установки, в соответствии с таблицей 9.

Таблица 9 — 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-	Актуально только для

		<env>.<public.domain>	Nutmeg и выше
mtadmin-<env>	CNAME	fsapi-<env>.<public.domain>	
webbackupapi-<env>	CNAME	fsapi-<env>.<public.domain>	

2.4 Сетевые настройки

Во время установки на все сервера будет установлен межсетевой экран **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 10-11.

Таблица 10 — Сетевые порты

	Связанный IP	**Порт**	**Протокол**	**Назначение**
Сервера роли fs_fe	0.0.0.0	80	TCP	HTTP-доступ к системе и API
	172.17.0.1	81		Мониторинг nginx
	0.0.0.0	443		HTTPS-доступ к системе и API
Сервера роли fs_be	172.17.0.1	26		SMTP для внутренних служб (выполняет перенавление на PSN)
	172.17.0.1	5432		Прокси stolon
	0.0.0.0	8080		HTTPS-доступ к системе и API
	172.17.0.1	8280		Прокси swift
	0.0.0.0	9090		mtadmin
	0.0.0.0	9091		web admin API
	0.0.0.0	9092		web API
	0.0.0.0	9095		card API
	0.0.0.0	17777		Прокси fs
Сервера роли fs_db	0.0.0.0	2379		Клиентский порт etcd
	0.0.0.0	2380		Кластерный порт etcd

	0.0.0.0	5432		Прокси stolon
	0.0.0.0	6379		Клиентский порт redis
	0.0.0.0	16379		Кластерный порт redis
	0.0.0.0	25432		stolon keeper
	0.0.0.0	26379		redis sentinel
Сервера роли fs_st	0.0.0.0	873		swift rsyncd
	0.0.0.0	6000		Сервер объектов swift
	0.0.0.0	6001		Сервер контейнеров swift
	0.0.0.0	6002		Сервер аккаунтов swift
	0.0.0.0	8280		Прокси-сервер swift
	0.0.0.0	11211		memcached
Сервера роли fs_infra	0.0.0.0	5000		docker registry
	0.0.0.0	8081		nginx (rpm-репозиторий)

Таблица 11 — 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, ESTABLISHED				ACCEPT
	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 Установка Хранилища

Данный дистрибутив предназначен для чистой установки либо для миграции с версии 2018.02 Mint. Информация по чистой установке приведена в п. 3.1-3.5. Информация по миграции описана в разделе 4.

3.1 Описание ролей

Роли, используемые при подготовке серверов, описаны в таблице 12.

Таблица 12 — Описание ролей, используемых при подготовке серверов

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_hosts1. **Важно!** Роль не работает на ОС Astra Linux!
resolv	Устанавливает hostname для выбранных серверов группы play_hosts2. **Важно!** Роль не работает на ОС Astra Linux!
selinux	Проверяет и запускает SELinux, а также активности режима enforcing*
yum	Настраивает пакетный менеджер yum, обновляет все пакеты до последней актуальной версии в подключенных репозиториях, за исключением kernel*, docker-ce*, container* *
package_tools	Добавляет в ОС репозитории, указанные в vars, дополнительно устанавливает требуемые пакеты по умолчанию, в т.ч. специфичные для поддерживаемых ОС
locale	Устанавливает выбранные параметры locale на сервера группы play_hosts
timezone	Устанавливает часовой пояс на сервера группы play_hosts
sshd	Настраивает конфигурационный файл sshd по шаблону с

* Только для RedHat-based ОС.

	указанными параметрами
chrony	Устанавливает и настраивает ntp (сервис синхронизации времени) на сервера группы play_hosts*
timesyncd	Устанавливает и настраивает сервис синхронизации времени**
sysctl	Устанавливает требуемые параметры ядра на сервера группы play_hosts
limits	Устанавливает ограничения на сервера группы play_hosts
kernel_ml	Устанавливает репозиторий elrepo последнего доступного ядра*
sensu	Устанавливает и настраивает клиентскую часть sensu, подключает стандартную подписку для мониторинга ресурсов сервера
rsyslog	Устанавливает и настраивает сервис сбора журналов ОС
docker_service	Устанавливает и настраивает сервис docker, подключает к развернутому docker_registry
iptables	Устанавливает и настраивает службы межсетевого экрана с параметрами, требуемыми для конкретной роли серверов

Роли, используемые для подготовки Хранилища, описаны в таблице 13.

Таблица 13 — Описание ролей, используемых для подготовки 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 сервис высокой доступности. **Важно!** Параметр используется только для отказоустойчивой установки
unbound	Устанавливает и настраивает кэширующий DNS.
etcd	Устанавливает кластер etcd
stolon на fs_db группу	Устанавливает и настраивает СУБД PostgreSQL с отказоустойчивым механизмом stolon

** Только для Astra Linux.

redis на fs_db группу	Устанавливает и настраивает Redis с отказоустойчивым механизмом sentinel
swift	Устанавливает и настраивает объектное хранилище Swift
stolon на fs_be группу	Устанавливает и настраивает прокси до СУБД PostgreSQL с отказоустойчивым механизмом stolon
redis на fs_be группу	Устанавливает и настраивает прокси Redis с отказоустойчивым механизмом sentinel
postfix	Устанавливает роль для развертывания MTA
fleet	Устанавливает и настраивает менеджер сервисов systemd, связанный с etcd в распределенную систему инициализации
fs (tag fs_setup)	Устанавливает и настраивает все необходимые сервисы fs
nginx	Устанавливает и настраивает веб-сервер nginx
confd	Устанавливает шаблоны и конфигурации для настроенных сервисов (в том числе и nginx)
fs (tag fs_accounts)	Создает необходимые учетные записи для приложений (co, psn, logos)

3.2 Порядок установки

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

Порядок установки Хранилища:

1. Подготовка серверов:
    1.1. Установка **docker_registry** и пакетного репозитория.
    1.2. Установка **ansible**.
    1.3. Установка системы.
2. Настройка параметров установки.
3. Запуск и проверка установки.

3.3 Подготовка серверов

3.3.1 Установка docker_registry и пакетного репозитория

Установка производится на сервере с ролью **fs_infra**.

Проверка: вход выполнен под пользователем **root**.

Этапы установки:

1. Доставить на сервер файл **fs_infra_*.run** путем монтирования ISO-образа или копирования файла на сервер по протоколу SFTP.
2. Запустить скрипт установки: bash fs_infra_[RELEASE].run, где RELEASE – имя релиза.
3. Принять приветственное уведомление.
4. Нажать на клавишу "Y" для продолжения установки.

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 **docker_registry** и пакетного репозитория будет завершена.

3.3.2 Установка ansible

Установка производится на рабочем месте оператора.

Проверка:

1. Вход выполнен под пользователем **root** или под пользователем sudo с привилегиями на **yum (dnf)**.
2. Машина соответствует требованиям, указанным в разделе 2 настоящей Инструкции.
3. С выбранного сервера есть возможность доступа по SSH до других серверов, на которых выполняется установка.
4. Ansible установлен, другие конфигурационные файлы ansible не присутствуют в системе.

Этапы установки:

1. Поместить ansible_bin_releaseXX.run в домашний каталог пользователя, из-под которого выполняется установка.
2. Запустить скрипт установки ansible_bin_releaseXX.run.
3. Принять приветственное уведомление.
4. Нажать "Y" для продолжения установки.

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 ansible будет завершена.

3.3.3 Установка системы с машины оператора

Этапы установки:

1. Перейти в каталог ~/install_fs/* с помощью команды:

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

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

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

3. Подготовить файл **inventory**. Примеры заполненных файлов можно найти в каталоге ~/contrib/fs/:
    1. файл hosts_cluster.yml – пример заполнения файла для отказоустойчивой установки;
    2. hosts_sa.yml – пример заполнения файла для установки без поддержки отказоустойчивости.
4. Копировать 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 **swift** (в части имени кластера), **sensu** (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 roles/ca/files/ пакета установки. Подробнее про размещение ключей можно прочитать в пп. 3.3.3.1.
7. Заполнить файл с параметрами, которые будут использоваться для проведения установки.
8. Создать в **group_vars** каталог для группы серверов с именем, которое было присвоено в файле **inventory**. Скопировать туда файл 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
```

9. Открыть файл из каталога размещения.

---

* Далее в отношении ДУ используются относительные пути.

10. Отредактировать значение параметров по комментариям. Примеры параметров для минимальной настройки можно найти в п. 3.4.1.

11. Включить на сервере роли **fs_infra** сервисы хранилища пакетов и docker registry. Для этого нужно воспользоваться командой:

```
[root@fs-infra-01 ~]# systemctl start fs-yumrepo
[root@fs-infra-01 ~]# systemctl start docker-registry.service
```

3.3.3.1 Размещение ssl-сертификатов для шифрования

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

Порядок размещения сертификатов:

1. Создать каталог:

```
[root@fs-installer ~]# mkdir roles/ca/files
```

2. Разместить сертификат внешнего домена:

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

3. Разместить ключ внешнего домена:

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

4. Разместить цепочку сертификатов промежуточных CA внешнего домена:

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

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

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

На следующем шаге при заполнении переменных, с использованием тестовых имен файлов, получатся переменные такого вида:

```
nginx:
  deployed_tls:
    - cert_filename: 'external_server.cert.pem'
      key_filename: 'external_server.key.pem'
```

```
      ca_filename: 'external_server.ca.pem'

postfix:
  tls:
    cert_filename: 'external_server.cert.pem'
    key_filename: 'external_server.key.pem'
    ca_filename: 'external_server.ca.pem'

fs:
  ssl:
    service:
      cert_filename: 'bundle_external_server.cert.pem'
      key_filename: 'stageoffice.ru-key.pem'
```

3.4 Настройка параметров установки

3.4.1 Минимальные параметры установки

Минимальные параметры, обязательные для заполнения:

- ansible_user;
- package_tools;
- docker;
- iptables;
- keepalived;
- unbound;
- etcd;
- stolon;
- redis;
- swift;
- postfix;
- fs;
- nginx.

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

3.4.1.1 Конфигурация параметров ОС

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

Таблица 14 — Конфигурация параметров ОС

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
ansible_user:	Имя пользователя для ssh-драйвера ansible	Str	admin
iptables:	Управление конфигурацией фаервола		
enabled:	Включение автоматической конфигурации IPTables. Рекомендуется оставить включенным (true).	Bool	True

3.4.1.2 Конфигурация параметров репозитория пакетов

Требуется указать адреса серверов с репозиторием пакетного менеджера при помощи следующих параметров:

Таблица 15 — Конфигурация параметров репозитория пакетов

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
package_tools:			
yum_repos:	**Только для CentOS:** Конфигурация yum-репозитория		
nct:	Указывается ссылка (URL) репозитория на инфраструктурном сервере (NCT) в формате: http://<*REPO*>:8081, где <*REPO*>- адрес сервера с ролью fs_infra	Str	нет

apt_repos:			Параметр настраивает APT-репозиторий. **Важно!** Заполняется, если установка выполняется на Alt или Astra Linux		
	"Altlinux":		AltLinux: Конфигурация APT-RPM репозитория		
		nct:			
			repo:	Указывается ссылка (URL) репозитория, в формате: ["rpm http://<*REPO*>:8083 x86_64 main"], где <REPO> - адрес сервера с ролью fs_infra	list	
	"Astra Linux (Orel)":		Astra Linux: Конфигурация APT-репозитория		
		nct:			
			repo:	Указывается ссылка (URL) репозитория в формате: ["deb http://<*REPO*>:8082 stretch main"], где <REPO> - адрес сервера с ролью fs_infra	list	
			gpg:	Указывается ссылка (URL) GPG-ключа в формате: http://<REPO>:8082/ pubkey.gpg, где <REPO> - адрес сервера с ролью fs_infra	str	

Примеры корректно настроенных параметров:

```
# CentOS
package_tools:
  yum_repos:
    nct: "http://ira01.example.org:8081"

# Astra Linux
package_tools:
  apt_repos:
    "Astra Linux (Orel)":
      nct:
        repo: ["deb http://ira01.example.org:8082 stretch main"]
        gpg: "http://ira01.example.org:8082/pubkey.gpg"

# Alt Linux
package_tools:
  apt_repos:
    “AltLinux”:
      repo: ["rpm http://ira01.example.org:8083 x86_64 main"]
```

3.4.1.3 Конфигурация docker

Для установки требуется указать адреса серверов с Docker Registry и DNS путем настройки следующих параметров:

Таблица 16 — Конфигурация docker

Параметр		**Описание**	**Тип**	**Значение по умолчанию**
docker:				
	image_tag:	Данное значение определяется автоматически по имени релиза. Например: если имя релиза MyOffice_FS_2019.02.RELEASE, тогда image_tag будет 2019.02.RELEASE		
	registry:			

		endpoint:	Указывается ссылка (URL) Docker Registry в формате: “<INFRA_HOSTNAME>:5000”, где <INFRA_HOSTNAME> - адрес машины с ролью fs_infra		
	daemon:		Параметры docker-демона		
		parameters:			
		"insecure-registries":	Список insecure-registries. Обязательно необходимо добавить registry с сервера с инфраструктурной ролью	list[str]	
		"dns":	Список dns-серверов для контейнеров. Рекомендуется использовать unbound (устанавливается на сервера с ролью fs_db или standalone)	list[str]	

Примеры корректно настроенных параметров:

```
docker:
  image_tag: "2019.02.RELEASE"
  registry:
    endpoint: "ira01.example.org:5000"

  daemon:
    parameters:
      "insecure-registries": ["ira01.example.org:5000"]
      "dns":
        # db01.example.org
        - “192.168.1.8”
        # db02.example.org
        - “192.168.1.9”
        # db03.example.org
        - “192.168.1.10”
```

3.4.1.4 Конфигурация keepalived

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

Для настройки параметра:

Таблица 17 — Конфигурация keepalived

Параметр **keepalived:**	**Описание**	**Тип**	**Значение по умолчанию**
vrrp:			
instances:			
fs_fe:	Конфигурация VIP для FSApi		
virtual_ip:	Указывается виртуальный ip-адрес		
password:	Указывается пароль		
fs_db:	Конфигурация VIP для БД PostgreSQL		
virtual_ip:	Указывается виртуальный ip-адрес.		
password:	Указывается пароль		

Примеры 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"
```

3.4.1.5 Конфигурация unbound

Данный параметр не рекомендуется отключать!

Для настройки параметра:

Таблица 18 — Конфигурация unbound

Параметр **unbound:**	**Описание**	**Тип**	**Значение по умолчанию**
access_control:	Указывается адрес для доступа к сети docker		
- "172.17.0.0/16"			
local_zone:			
- 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:			
type:			
ip:			

Примеры корректно настроенных параметров:

```
unbound:
  access_control:
    - "172.17.0.0/16"

  local_zone:
    - type: "transparent""
      zone: ""
```

```
    local_data:
# Пример заполнения параметров при отказоустойчивой установке
      - 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"
# Пример заполнения параметров при установке 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"
```

3.4.1.6 Конфигурация хранилищ

Для настройки хранилищ:

Таблица 19 — Конфигурация хранилищ

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
etcd:			
token:	Указывается разделяемый секрет между экземплярами кластера, требуется для установки соединения. Например: «fs_42_3»		
stolon:			
postgres_password:	Конфигурация СУБД PostgreSQL		
replication:	Пароль пользователя postgres		
password:	Конфигурация репликации		
monitoring:	Пароль пользователя replication		
password:	Конфигурация мониторинга		
users:	Пароль пользователя monitoring		

		- name:			
		password:			
	pg_parameters:		Список параметров PostgreSQL – оптимизация производительности		
		max_connections:			
		shared_buffers:			
		effective_cache_size:			
		maintenance_work_mem:			
		checkpoint_completion_target:			
		wal_buffers:			
		default_statistics_target:			
		random_page_cost:			
		effective_io_concurrency:			
		work_mem:			
		min_wal_size:			
		max_wal_size:			
		max_worker_processes:			
redis:			Конфигурация REDIS		
	cluster:		Конфигурация кластера		
		master:	Номер master-ноды redis, например, “01”	Str	
		node_number:	Hostname сервера, на котором будет размещен redis. Должен соответствовать hostname в inventory		
		hosting_server:	Пароль для доступа к		

				redis		
			password:	Список slave (для отказоустойчивой инсталляции)		
		slaves:		Номер master-ноды redis, например, “02”		
			- node_number:	Hostname сервера, на котором будет размещен redis. Должен соответствовать hostname в inventory		
			hosting_server:	Hostname сервера, на котором будет размещен redis. Должен соответствовать hostname в inventory		
	sentinel:			Конфигурация redis-sentinel		
		cluster:		Список узлов кластера. Должен включать в себя список slave-узлов и master		
			- node_number:			
			hosting_server:			
swift:				Конфигурация хранилища swift		
	cluster:			Конфигурация кластера		
		nodes:		Список узлов кластера		
			- node_number:	Номер узла как строка. Например, “01”		
			hosting_server:	Hostname сервера, на котором будет размещен узел. Должен соответствовать hostname в inventory		
			drives:	Список дисков для хранения. Диск должен быть примонтирован в директорию */srv/node/<drive_name>*		
	users:			Конфигурация		

			пользователей		
		fs:			
			key:	Пароль пользователя FS		

Примеры корректно настроенных параметров:

```
etcd:
  token: ""
  cluster:
# Пример заполнения параметров при отказоустойчивой установке
    - "etcd.db01.installation.example.net"
    - "etcd.db02.installation.example.net"
    - "etcd.db03.installation.example.net"

# Пример заполнения параметров при установке без поддержки
отказоустойчивости

    - 'etcd.sa01.installation.example.net'

stolon:

  replication:
    password: ""

  monitoring:
    password: ""

  users:
    - name: "fs"
      password: ""
    - name: "fs_search"
      password: ""

# Пример заполнения ниже рассчитан на16 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 при установке без поддержки отказоустойчивости

    master:
      node_number: "01"
      hosting_server: ""
      password: ""
# Пример заполнения параметров при отказоустойчивой установке
    slaves:
      - node_number: "02"
        hosting_server: "" # db02.installation.example.net
      - node_number: "03"
        hosting_server: "" # db03.installation.example.net

  sentinel:

# Пример заполнения параметров при установке без поддержки отказоустойчивости

    cluster:
      - node_number: "01"
        hosting_server: ""
# Пример заполнения параметров при отказоустойчивой установке
      - node_number: "02"
        hosting_server: "" # db02.installation.example.net
      - node_number: "03"
        hosting_server: "" # db03.installation.example.net

swift:
  cluster:

# Пример заполнения параметров при установке без поддержки отказоустойчивости

    nodes:
      - node_number: "01"
```

```
      hosting_server: ""
      drives:
# Диск /srv/node/data0
        - "data0"
# Пример заполнения параметров при отказоустойчивой установке
    - 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"
  users:
    fs:
      # Password key for swift account FS
      # Generate it with `pwgen 32`
      key: ""
```

3.4.1.7 Конфигурация сервиса передачи электронной почты

Для настройки сервиса передачи электронной почты:

Таблица 20 — Пример заполнения параметров без поддержки отказоустойчивости

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
postfix:			
relay:			
relayhost:	Указывается hostname почтового релея (PSN)		psn-sa-1.installation.example.net или smtp.example.ne
name:			
parameters:			
default:			

				mydomain:	Указывается домен по умолчанию для установки		
		tls:					
			cert_filename:		Указывается имя файла TLS-сертификата		
			key_filename:		Указывается имя файла ключа от TLS-сертификата		
			ca_filename:		Указывается имя файла с корневым CA		

Примеры корректно настроенных параметров:

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

3.4.1.8 Конфигурация fs

При миграции с предыдущих релизов FS требуется использовать те же значения параметров salt1-salt6, salt_no_captcha и salt_no_auth, что и использовались ранее. В противном случае пользователи инсталляции не смогут авторизоваться

При установке без поддержки отказоустойчивости задается параметр cluster: False.

Для настройки Хранилища:

Таблица 21 — Конфигурация fs

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
fs:	Конфигурация сервиса FS		
cluster:	Отказоустойчивая конфигурация	bool	true
server_id:	ID сервера. Строка с номером инсталляции	str	“777”
default_domain:	Домен инсталляции	str	
domain_module:	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:		list[str]	
esia:	Конфигурация интеграции с ЕСИА		
enabled:	Включить интеграцию	bool	false
psn:	Конфигурация интеграции с PSN		
enabled:	Включить интеграцию	bool	false
kerberos:	Конфигурация интеграции с AD		
enabled:	Включить интеграцию с AD	bool	false
default_realm:	Kerberos Realm по умолчанию	str	
realms:	Список Kerberos Realms	list[dict]	
- name:	Имя домена	str	
controllers:	Список контроллеров домена	list[str]	

tenants:			Конфигурация тенантов		
	default:		Конфигурация тенанта по умолчанию		
		default_domain:	Имя домена	str	
		recovery_email:	E-mail для восстановления пароля	str	
		admin:			
		password:	Пароль администратора	str	
ssl:			Конфигурация сертификатов для HTTPS		
	service:				
		cert_filename:	Имя сертификата	str	
		key_filename:	Имя ключа	str	
salts:			Конфигурация солей шифрования		
	shared:		Соль шифрования внутренних запросов. Генерируется при помощи команды **pwgen -s 16**	str	
	salt1:		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:		Параметр СО **FS_TOKEN_SALT**	str	
	salt_no_auth:		Параметр СО **FS_TOKEN_SALT_EXT**	str	
crypt:			Конфигурация криптографических ключей (Интеграция с СО)		
	main:				

			aes_iv:	Параметр СО **FS_APP_ENCRYPTION_IV**	str	
			aes_key:	Параметр СО **FS_APP_ENCRYPTION_KEY**	str	
		session:				
			aes_iv:	Параметр СО **AUTH_ENCRYPTION_IV**	str	
			aes_key:	Параметр СО **AUTH_ENCRYPTION_KEY**	str	
			aes_salt:	Параметр СО **AUTH_ENCRYPTION_SALT**	str	
	swift:			Конфигурация доступа к Swift		
		account:		Имя аккаунта в Swift	str	fs
		username:		Имя пользователя в Swift	str	fs
		password:		Пароль пользователя в Swift (**swift.users.fs.key**)	str	
	redis:			Конфигурация доступа к redis		
		password:		Пароль для доступа к redis (**redis.cluster.master.password**)	str	
		sentinels:		Список Redis Sentinels	list[str]	
	databases:			Конфигурация доступа к СУБД		
		main:		Конфигурация БД fs		
			username:	Имя пользователя	str	
			password:	Пароль	str	
			database:	Имя БД	str	
		search:		Конфигурация БД search		

		username:	Имя пользователя	str	
		password:	Пароль	str	
		database:	Имя БД	str	
mtadminapi_user:			Конфигурация доступа к MTAdminAPI		
	password:		Пароль администратора	str	
appapi:			Конфигурация AppAPI	Конфигурация AppAPI	
	log:		Конфигурация логирования		
		level:	Уровень логирования	str	
	accounts:		Конфигурация APP-учетных записей		
		co:	Конфигурация УЗ app-co		
		password:	Пароль УЗ app-co	str	
		psn:	Конфигурация УЗ app-psn		
		password:	Пароль УЗ app-psn	str	
		logos:	Конфигурация УЗ app-logos		
		password:	Пароль УЗ app-logos	str	
password_policy:			Конфигурация парольной политики		
	min_length		Минимальная длина пароля	int	8
	min_numbers		Минимальное количество цифр в пароле	int	1
	min_special_symbols		Минимальное количество специальных символов в пароле	int	0
	letters_required		Требовать наличия букв в пароле	bool	True
	mixed_case_required		Требовать наличия букв в разном регистре	bool	False
	default_expire_time		Время истечения срока действия пароля	int	0

captcha:		Конфигурация captcha		
	fail_captcha_start	Количество попыток неправильного ввода пароля для включения captcha	int	3
	fail_ip_skip	Whitelist для IP-адреса клиента	str	“10.20.”
	fail_lifetime	Время, которое клиенту будет требоваться ввести captcha	int	3600
	fail_max_attempts	Максимальное количество попыток неправильного ввода пароля для отключения авторизации	int	8
	fail_timeout	Время, на которое отключается авторизация	int	300
ratelimit:				
	enabled	Включить ratelimit	bool	True
	bruteforce_attempts	Количество попыток подбора пароля для включения captcha	int	3000
	bruteforce_period	Период, в который должно состояться включение captcha	int	1000
	ddos_attempts	Количество запросов с одного адреса для включения captcha	int	6000
	ddos_period	Период, в который должно состояться включение captcha	int	3000

3.4.1.9 Конфигурация развертывания веб-сервера

Для настройки параметра:

Таблица 22 — Конфигурация развертывания веб-сервера

Параметр **nginx:**	**Описание**	**Тип**	**Значение по умолчанию**
deployed_tls:			
- cert_filename:			
key_filename:			
ca_filename:			

Примеры корректно настроенных параметров:

```
nginx:
  deployed_tls:
# Указывается имя сертификата. Важно! Сертификат должен располагаться в каталоге roles/ca/files
    - cert_filename: ""
# Указывается имя сертификата. Важно! Сертификат должен располагаться в каталоге roles/ca/files
      key_filename: ""
# Указывается имя сертификата. Важно! Сертификат должен располагаться в каталоге roles/ca/files
      ca_filename: ""
```

3.4.2 Настройка без отказоустойчивости

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

При выполнении установки без поддержки отказоустойчивости дополнительно потребуется заполнить параметры, указанные в таблице 23.

Таблица 23 — Заполнение параметров при установке

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"

3.4.3 Отказоустойчивая установка

При выполнении отказоустойчивой установки дополнительно потребуется заполнить параметры, указанные в таблице 24.

Таблица 24 — Заполнение параметров при отказоустойчивой установке

Для параметра Swift	**Параметр**	**Пример заполнения**
	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"

3.5 Запуск и проверка установки

3.5.1 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 13.

2. Запустить команду на установку FS:

```
[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 сервиса высокой доступности (только в случае установки на кластер);
- unbound – установка и настройка кеширующего DNS (ускоряет работу внутри стенда установки, создаёт возможность работы на серверных IP);
- etcd – установка и настройка распределенного хранилища key-value;
- stolon на fs_db группу – установка и настройка СУБД Postgres с отказоустойчивым механизмом stolon;
- redis на fs_db группу – установка и настройка хранилища структуры данных redis с отказоустойчивым механизмом sentinel;
- swift – установка и настройка объектного хранилища Swift;
- stolon на fs_be группу – установка и настройка proxy до СУБД Postgres с отказоустойчивым механизмом stolon;
- redis на fs_be группу – установка и настройка proxy redis с отказоустойчивым механизмом sentinel;
- postfix – установка и настройка MTA;

- fleet – установка и настройка менеджера сервисов systemd, связанный с etcd в распределенную систему инициализации;
- fs (tag fs_setup) – установка и настройка всех требуемых сервисов fs;
- 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;
- fs (tag fs_accounts) – создание учётных записей co, psn, logos для приложений.

3.5.2 Проверка 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.

4 Миграция данных

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 объектного-хранилища Swift;
- Распределенное хранилища 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 систему в процессе резервного копирования.

4.1 Резервная копия 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.

4.1.1 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 приведён пример имён 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 ФС
```

Имя БД Logos можно посмотреть на серверах с ролью **logos**, в параметрах запущенного процесса сервиса **baham**. Для вывода параметров в списке процессов 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/db_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/db_lgs?sslmode=disable&search_path=catalog,extensions
```

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

4.1.2 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):

```
PGPASSWORD="fs_search_password" pg_dump -U fs_search db_search_user | ssh me@backupserver "cat > /backup/fs_oldrelease/fs_search.sql"
PGPASSWORD="fs_vmail_password" pg_dump -U fs_vmail db_vmail_user | ssh me@backupserver "cat > /backup/fs_oldrelease/fs_vmail.sql"
PGPASSWORD="fs_password" pg_dump -U fs db_fs_user | ssh me@backupserver "cat > /backup/fs_oldrelease/fs.sql"
PGPASSWORD="lgs_password" pg_dump -U lgs db_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, осуществяющая 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 Резервная копия объектного хранилища Swift

4.2.1 Установка rclone

Утилита **rclone** распространяется по лицензии MIT и предназначается для резервного копирования объектного хранилища **Swift**.

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

4.2.2 Настройка rclone

Настройка утилиты на работу с текущим экземпляром объектного хранилища Swift 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 объектному хранилищу;
- user – "имя аккаунта : имя пользователя";
- key – ключ учётной записи;
- auth – путь до прокси на объектное хранилище **Swift**. При этом нужно поменять IP-адрес и порт на необходимые;
- storage_url – путь до прокси на объектное хранилище **Swift**. 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 акаунта 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.

4.2.3 Выполнение резервного копирования объектного хранилища Swift

Для резервного копирования объектного хранилища **Swift**:

1. Создать каталог для сохранения всех объектов **Swift** на сервере резервного копирования, перейти в каталог.

```
[root@st1 ~]# mkdir /backup/fs_oldrelease/swift/
[root@st1 ~]# cd /backup/fs_oldrelease/swift/
```

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

```
[root@st1 ~]# for i in $(rclone lsf fs:); do rclone sync --progress fs:${i} ${i}; done
```

4.3 Резервная копия распределённого хранилища etcd

Распределённое хранилище **etcd** хранит данные и ключи дополнительных доменов, поэтому требуется создать его резервную копию. Для этого:

1. Выполнить команду для создания слепка всего хранилища на серверах с ролью **database**:

```
[root@db1 ~]# ETCDCTL_API=3 etcdctl snapshot save /tmp/snapshot.db
```

2. Перенести файл слепка на сервер резервного копирования:

```
[root@db1 ~]# scp /tmp/snapshot.db me@backupserver:/backup/fs_oldrelease/
```

4.4 Резервная копия важных переменных для миграции

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

Ниже приведен перечень переменных, которые необходимо сохранить, и места, где их можно найти:

- токен **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 системы 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
```

- учётная запись к объектному хранилищу **Swift** – 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
```

5 Дополнительное

5.1 Расширенная настройка

Далее приведены параметры, которые могут быть настроены дополнительно.

5.1.1 Параметр «chrony»

Назначение параметра: настройка серверного времени.

Таблица 25 — Параметр «chrony»

Параметр **chrony:**	**Описание**	**Тип**	**Значение по умолчанию**
ntp:			
servers:			

Примеры корректно настроенных параметров:

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

```
- server1.ltd
```

5.1.2 Параметр «confd»

Назначение параметра: настройка шаблонов для настроенных сервисов.

Таблица 26 — Параметр «confd»

Параметр **confd:**	**Описание**	**Тип**	**Значение по умолчанию**
pattern:			
conf_dir:			
backend:			
debug:			
prefix:			
watch:			
interval:			
resources:			
- service_name:			
src:			
dest:			
uid:			
gid:			
mode:			
key:			
check_cmd:			
reload_cmd:			
servers:			

		- hosting_server:			
		resources:			
			- service_name:		
			src:		
			dest:		
			uid:		
			gid:		
			mode:		
			key:		
			check_cmd:		
			reload_cmd:		

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"
    - service_name: "apache"
      src: "apache.tmpl"
      dest: "/etc/apache/sites-enabled/<appname>.conf"
```

```
    uid: 0
    gid: 0
    mode: "0644"
    key:
      - "/services/"
      - "/nginx"
    check_cmd: "/usr/sbin/apache -t"
    reload_cmd: "/usr/sbin/service apache reload"
servers:
  - hosting_server: "confd-srv1.example.com"
    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"
      - service_name: "apache"
        src: "apache.tmpl"
        dest: "/etc/apache/sites-enabled/<appname>.conf"
        uid: 0
        gid: 0
        mode: "0644"
        key:
          - "/services/"
          - "/nginx"
        check_cmd: "/usr/sbin/apache -t"
        reload_cmd: "/usr/sbin/service apache reload"
  - hosting_server: "confd-srv2.example.com"
    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"
  - service_name: "apache"
    src: "apache.tmpl"
    dest: "/etc/apache/sites-enabled/<appname>.conf"
    uid: 0
    gid: 0
    mode: "0644"
    key:
      - "/services/"
      - "/nginx"
    check_cmd: "/usr/sbin/apache -t"
    reload_cmd: "/usr/sbin/service apache reload"
```

5.1.3 Параметр «docker_service»

Назначение параметра: настройка параметров работы Docker.

Таблица 27 — Параметр «docker_service»

Параметр	Описание	Тип	Значение по умолчанию
docker_service:			
repo_params:			
yum:			
url:			
gpgkey:			
apt:			
url:			
gpgkey:			
docker:			
daemon:			
parameters:			
use_tls:			

		tls:			
			algorithm:		
			key_size:		
		hosts:			
		cleaner:			
			enabled:		
			minute:		
			hour:		
			day:		
			month:		
			weekday:		
			user:		

Примеры корректно настроенных параметров:

```
docker_service:
  repo_params:
    yum:
      url: "https://download.docker.com/linux/centos/7/$basearch/stable"
      gpgkey: "https://download.docker.com/linux/centos/gpg"
    apt:
      url: "deb [arch=amd64] https://download.docker.com/linux/debian stretch stable"
      gpgkey: "https://download.docker.com/linux/debian/gpg"

docker:
  daemon:
    parameters: {}

    use_tls: False

    tls:
      algorithm: "ecdsa"
      key_size: 256

    hosts:
      - "unix:///var/run/docker.sock"
      - "tcp://127.0.0.1:2375"
  cleaner:
```

```
enabled: False
minute: "10"
hour: "23"
day: "*"
month: "*"
weekday: "*"
user: "root"
```

5.1.4 Параметр «etcd»

Назначение параметра: настройка параметров работы etcd.

Таблица 28 — Параметр «etcd»

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
etcd:			
auth:			
password:			
cluster:			
use_tls:	При значении «True» будут созданы tls-сертификаты и ключи		
image:			
registry:			
name:			
tag:			
settings:			
heartbeat_interval:			
election_timeout:			
snapshot_count:			
quota_backend_bytes:			
max_snapshots:			

		max_wals:			
		enable_v2:			
		cert_auth:			
			peer:		
			client:		
	token:				

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"
```

5.1.5 Параметр «fleet»

Назначение параметра: настройка подсистемы распределённого управления контейнерами.

Таблица 29 — Параметр «fleet»

Параметр **fleet:**	**Описание**	**Тип**	**Значение по умолчанию**
etcd			
request_timeout:			
reconcile_interval:			
agent_ttl:			
use_tls:			
use_auth:			

Примеры корректно настроенных параметров:

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

5.1.6 Параметр «fs»

Назначение параметра: настройка установки Хранилища.

Таблица 30 — Параметр «fs»

Параметр **fs:**	**Описание**	**Тип**	**Значение по умолчанию**
cluster:	Значение False указывается в случае установки на оборудование без отказоустойчивости		

image:				
	registry:			
	fsapi_name:			
	appapi_name:			
	cron_name:			
	indexer_name:			
	tag:			
ports:				
	appapi:			
	cardapi:			
	mtadmin:			
	proxy:			
	webadminapi:			
	webapi:			
server_id:				
domain_module:				
default_domain:				
allowed_domains:				
esia:				
	enabled:			
psn:				
	enabled:			
kerberos:				
	enabled:			
	default_realm:			
	realms:			
		name:		
		controllers:		

webapi:				
	debug:			
	custom:			
indexer:				
	worker:			
	delay:			
	reconn:			
	lang:			
	swift:			
		username:			
appapi:				
	redis_pool_size:			
	postgres_pool_size:			
	worker_count:			
	session:			
		timeout:			
	log:			
		level:			
		facility:			
	accounts:			
		logos:			
			password:			
		co:			
			password:			
		psn:			
			password:			
mtadmin:				
	custom:			

	uwsgi:				
		plugins:			
		services:			
	password_policy:				
		default_expire_time:			
		min_length:			
		min_numbers:			
		min_special_symbols:			
		letters_required:			
		mixed_case_required:			
	ratelimit:				
		enabled:			
		bruteforce_attempts:			
		bruteforce_period:			
		ddos_attempts:			
		ddos_period:			
	captcha:				
		fail_captcha_start:			
		fail_ip_skip:			
		fail_lifetime:			
		fail_max_attempts:			
		fail_timeout:			
	tenants:				
		default:			
			default_domain:		
			recovery_email:		
			tariff_id:		
			lang:		

			admin:			
				username:		
				password:		
				firstname:		
				middlename:		
				lastname:		
		example_com:				
			default_domain:			
			recovery_email:			
			admin:			
				username:		
				password:		
	subdomains:					
	crypt:					
		main:				
			aes_iv:			
			aes_key:			
		session:				
			aes_iv:			
			aes_key:			
			aes_salt:			
	salts:					
		shared:				
		salt1:				
		salt2:				
		salt3:				
		salt4:				
		salt5:				

	salt6:			
	salt_no_captcha:			
	salt_no_auth:			
swift:				
	account:			
	username:			
	password:			
databases:				
	use_prepared_statements:			
	main:			
	username:			
	password:			
	database:			
	search:			
	username:			
	password:			
	database:			
redis:				
	password:			

Примеры корректно настроенных параметров:

```
fs:
  cluster: False

  image:
    registry: "{{ docker.registry.endpoint }}"
    fsapi_name: "nct_fs"
    appapi_name: "nct_appapi"
    cron_name: "nct_fs_cron"
    indexer_name: "nct_indexer"
    tag: "latest"
```

```
ports:
  appapi: 8080
  cardapi: 9095
  mtadmin: 9090
  proxy: 17777
  webadminapi: 9091
  webapi: 9092

server_id: "1"

domain_module: "{service}-{domain}"

default_domain: "example.com"

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"

webapi:
  # Enable debugging of FsAPI services
  debug: False
  # Extra FsAPI Configuration Directives
  custom: {}

indexer:
  worker: 2
  delay: 3
  reconn: 2
  lang: "russian_email"
```

```
  swift:
    username: "admin"

appapi:
  redis_pool_size: 1
  postgres_pool_size: 10
  worker_count: 2
  # Session Configuration
  session:
    timeout: 86400
  log:
    level: "info"
    facility: "local6"
  accounts:
    logos:
      password: "changeme12345"
    co:
      password: "changeme12345"
    psn:
      password: "changeme12345"

mtadmin:
  custom: {}

uwsgi:
  plugins:
    - "http"
    - "psgi"
    - "syslog"
  services:
    - "cardapi"
    - "mtadmin"
    - "proxy"
    - "webadminapi"
    - "webapi"

password_policy:
  default_expire_time: 0
  min_length: 8
  min_numbers: 1
  min_special_symbols: 0
  letters_required: True
```

```
  mixed_case_required: False

ratelimit:
  enabled: True

  bruteforce_attempts: 3000
  bruteforce_period: 1000
  ddos_attempts: 6000
  ddos_period: 3000

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

tenants:
  default:
    default_domain: "example.com"
    recovery_email: "admin@example.com"
    tariff_id: "corp_1"
    lang: "ru-RU"

    admin:
      username: "fsadmin"
      password: "changeme"
      firstname: "Admin"
      middlename: "A"
      lastname: "Admin"

  example_com:
    default_domain: "example.com"
    recovery_email: "admin@example.com"
    admin:
      username: "fsadmin"
      password: "changeme"

subdomains:
  - "admin"
  - "apidocs"
  - "appapi"
```

```
  - "auth"
  - "autoconfig"
  - "cab"
  - "cardapi"
  - "cdn"
  - "coapi"
  - "docs"
  - "files"
  - "fsapi"
  - "imap"
  - "links"
  - "logos "
  - "mail"
  - "mailapi"
  - "mtadmin"
  - "proxy"
  - "psnapi"
  - "push"
  - "smtp"
  - "viewer"

crypt:
  main:
    aes_iv:
    aes_key:
  session:
    aes_iv:
    aes_key:
    aes_salt:

salts:
  shared:
  salt1:
  salt2:
  salt3:
  salt4:
  salt5:
  salt6:

  salt_no_captcha:

  salt_no_auth:
```

```
  swift:
    account:
    username:
    password:
s
  databases:
    use_prepared_statements: True
    main:
      username:
      password:
      database:
    search:
      username:
      password:
      database:

  redis:
    password:
```

5.1.7 Параметр «hostname»

Назначение параметра: создание словаря в host_vars для определения hostname в /etc/hosts.

Таблица 31 — Параметр «hostname»

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
system:			
hostname:			

Примеры корректно настроенных параметров:

```
system:
  hostname: 'host.domain.ltd'
```

5.1.8 Параметр «iptables»

Назначение параметра: настройка политику для сервера.

Таблица 32 — Параметр «iptables»

Параметр	Описание	Тип	Значение по умолчанию
iptables:			
enabled:			
ssh_port:			
policy:			
input:			
output:			
forward:			
custom_ruleset:			

Примеры корректно настроенных параметров:

```
iptables:
  enabled: False
  ssh_port: 22
  policy:
    input: "DROP"
    output: "ACCEPT"
    forward: "DROP"
  custom_ruleset: []
```

5.1.9 Параметр «keepalived»

Назначение параметра: настройка keepalived.

Таблица 33 — Параметр «keepalived»

Параметр	Описание	Тип	Значение по умолчанию
keepalived:	Приоритет настроек отдается play_hosts. **Важно!** Не запускать при ограничении использования одного узла		
vrrp:			

		instances:			
			fe_vip:		
				virtual_ip:	
				router_id:	
				password:	
				unicast_peers:	

Примеры корректно настроенных параметров:

```
keepalived:
  vrrp:
    instances:
      fe_vip:
        virtual_ip: "192.168.1.1"
        router_id: 66
        password: "changeme"
        interface: "eth0"
        unicast_peers:
          - "192.168.1.10"
          - "192.168.1.11"
          - "192.168.1.12"
```

5.1.10 Параметр «kernel_ml»

Назначение параметра: включение elrepo, удаление старой версии kernel и установка новой.

Таблица 34 — Параметр «kernel_ml»

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
kernel_ml:			
elrepo_repo_url:			
elrepo_key_url:			

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"
```

5.1.11 Параметр «limits»

Назначение параметра: настройка ограничений для ОС.

Таблица 35 — Параметр «limits»

Параметр **limits:**	**Описание**	**Тип**	**Значение по умолчанию**
default_options:			
- domain:			
type:			
item:			
value:			
filename:			

Примеры корректно настроенных параметров:

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

5.1.12 Параметр «locale»

Назначение параметра: настройка языка и кодировки для каталогов.

Таблица 36 — Параметр «locale»

Параметр **locale:**	**Описание**	**Тип**	**Значение по умолчанию**
system:			
locales:			

Примеры корректно настроенных параметров:

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

5.1.13 Параметр «nginx»

Назначение параметра: настройка nginx.

Таблица 37 — Параметр «nginx»

Параметр **nginx:**	**Описание**	**Тип**	**Значение по умолчанию**
image:			
name:			
registry:			
tag:			
container:			
ports:			
conf:			
worker_processes:			
worker_connections:			
worker_rlimit_nofile:			
use:			
sendfile:			
tcp_nopush:			
tcp_nodelay:			
server_tokens:			
gzip:			
keepalive_timeout:			
client_max_body_size:			

		server_names_hash_bucket_size:			
	logs:				
		error:			
			path:		
			level:		
		access:			
			path:		
			format:		
	deployed_tls:				
		- cert_filename:			
		key_filename:			
		ca_filename:			
	ldap:				
		enabled:			
		image:			
			name:		
			tag:		
		ports:			
		bind_dn:			
			user:		
	vhosts:				
		server_name:			
		template:			
		upstream:			
		listen:			
		ssl:			
			redirect_to_https:		

			cert:			
			key:			

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"
```

5.1.14 Параметр «package_tools»

Назначение параметра: установка специальных пакетов для определенных ОС.

Таблица 38 — Параметр «package_tools»

Параметр	Описание	Тип	Значение по умолчанию
package_tools:			
repo_packages:			
apt_repos:			
your_custom:			
repo:			
gpg:			
yum_repos:			
your_custom:			
packages:			

Примеры корректно настроенных параметров:

```
package_tools:
 repo_packages:
  'https://host.domain.ltd/repository/repository_name/package.rpm': 'latest'
 apt_repos:
  'Altlinux':
   your_custom:
    repo: ['rpm https://host.domain.ltd/repository/repository_name/ x86_64 main']
  'Astra Linux (Orel)':
   your_custom:
    repo: ['deb https://host.domain.ltd/repository/repository_name/ stretch main']
    gpg: 'https://host.domain.ltd/repository/pubkey.gpg'
 yum_repos:
  your_custom: "https://host.domain.ltd/"
 packages: {}
```

5.1.15 Параметр «postfix»

Назначение параметра: настройка postfix.

Таблица 39 — Параметр «postfix»

Параметр **postfix:**	**Описание**	**Тип**	**Значение по умолчанию**
image:			
registry:			
tag:			
name:			
uid:			
gid:			
relay:			
container:			
bind_ip:			
bind_port:			25
log_driver:			

			log_options:			
				syslog-address:		
				tag:		
		use_tls:				
		tls_mode:				
		tls:				
			cert_filename:			
			key_filename:			
			ca_filename:			
		relayhost:				
			name:			
			port:			25
		milter:				
			enabled:			
			endpoints:			
		parameters:				
			custom:			
			default:			
				mydomain:		
				relay_domains:		
				mynetworks:		
				restrictions:		
					client:	
					helo:	
					sender:	
					recipient:	
				limits:		

					anvil_rate_time_unit:			
					anvil_status_update_time:			
					bounce_size_limit:			
					default_process_limit:			
					lmtp_destination_concurrency_limit:			
					max_use:			
					message_size_limit:			
					smtpd_client_message_rate_limit:			
					smtpd_client_connection_count_limit:			
	mx:					Конфигурация роли mx		
		use_tls:						
		use_woof_ldap:						
		milter:						
			enabled:					False
		parameters:						
			custom:					
			default:					
				relay_domains:				
				mynetworks:				
				restrictions:				
					client:			

					helo:			
					sender:			
					recipient:			
				limits:				
					anvil_rate_time_unit:			
					anvil_status_update_time:			
					bounce_size_limit:			
					default_process_limit:			
					lmtp_destination_concurrency_limit:			
					max_use:			
					message_size_limit:			
					smtpd_client_message_rate_limit:			
					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
```

5.1.16 Параметр «redis»

Назначение параметра: настройка redis.

Таблица 40 — Параметр «redis»

Параметр **redis:**	**Описание**	**Тип**	**Значение по умолчанию**
cluster:			
name:	Указывается наименование кластера		
master:			
password:			
node_number:			
hosting_server:			
slaves:			
- node_number:			
hosting_server:			
client_port:			
sentinel:			
quorum_number:			
down_after_ms:			

		failover_timeout:			
		parallel_syncs:			
		cluster:			
		- node_number:			
		hosting_server:			
		client_port:			
	tunnel:				
		nodes:			
		- node_number:			
		hosting_server:			
		databases:			
		name:			
		local_port:			

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
     - 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"
```

5.1.17 Параметр «resolv»

Назначение параметра: установка hostname.

Таблица 41 — Параметр «resolv»

Параметр **resolv:**	**Описание**	**Тип**	**Значение по умолчанию**
search:			
domain:	Указывается IP-адрес		
nameserver:			
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"
```

5.1.18 Параметр «rsyslog»

Назначение роли: настройка rsyslog для работы с docker.

Таблица 42 — Параметр «rsyslog»

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
rsyslog:			
docker:			
enabled:			
docker_socket:			
container_socket:			
forward:			
enabled:			
template:			
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"
```

5.1.19 Параметр «selinux»

Назначение параметра: настройка selinux по умолчанию.

Таблица 43 — Параметр «selinux»

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

Примеры корректно настроенных параметров:

```
selinux:
  # Choose default SELinux mode: enforcing, permissive, disabled
  state: enforcing
```

5.1.20 Параметр «ssh_keys»

Назначение параметра: настройка ssh-ключей.

Таблица 44 — Параметр «ssh_keys»

Параметр	Описание	Тип	Значение по умолчанию
ssh_keys:			
system_users:			
- name:			
authorized:			
- name:			
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"
```

5.1.21 Параметр «sshd»

Назначение параметра: настройка конфигурационного файла sshd.

Таблица 45 — Параметр «sshd»

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
ansible_port:			
sshd:			
protocol:			2
accept_env:			
permit_root_login:			yes
password_authentication:			yes
use_dns:			yes
x11_forwarding:			no
allow_groups:			
allow_users:			

Примеры корректно настроенных параметров:

```
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: []
```

5.1.22 Параметр «stolon»

Назначение параметра: настройка stolon.

Таблица 46 — Параметр «stolon»

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
stolon:			
cluster_name:	Наименование кластера. Должно быть уникальным		
etcd_endpoints:			
postgres_password:			
replication:	Конфигурация копий		
port:			
username:			
password:			
proxy:	Конфигурация прокси		
port:			
image:			
registry:			
tag:			
postgres_uid:			
postgres_gid:			
users:			
name:			
password:			
databases:			

		name:			
		owner:			

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"
```

5.1.23 Параметр «swift»

Назначение параметра: настройка swift.

Таблица 47 — Параметр «swift»

Параметр	Описание	Тип	Значение по умолчанию
swift:			
uid:			
gid:			
image:			
registry:			
name:			
tag:			
ring:			
part_power:			
replicas:	Количество копий		
min_part_hours:			
node_dir:			
cluster:			
name:			
nodes:			
- node_number:			
hosting_server:			
drives:			
ports:			
replication:			
ports:	Конфигурация порта		
account_server:			
container_server:			

		object_server:			
		replication:			
		proxy_server:			
		memcached:			
	users:				
		- account:			
		user:			
		key:			
		groups:			

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"
```

5.1.24 Параметр «sysctl»

Назначение параметра: настройка параметров kernel.

Таблица 48 — Параметр «sysctl»

Параметр	**Описание**	**Тип**	**Значение по умолчанию**
sysctl:			
default_sysctl_options:			
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
```

5.1.25 Параметр «timesyncd»

Назначение параметра: настройка синхронизации времени.

Таблица 49 — Параметр «timesyncd»

Параметр **timesyncd:**	**Описание**	**Тип**	**Значение по умолчанию**
- name:			
systemd:			
name:			
state:			
enabled:			
masked:			

Примеры корректно настроенных параметров:

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

5.1.26 Параметр «timezone»

Назначение параметра: настройка часового пояса для сервера.

Таблица 50 — Параметр «timezone»

Параметр **timezone:**	**Описание**	**Тип**	**Значение по умолчанию**
system:			
timezone:			

Примеры корректно настроенных параметров:

```
system:
```

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

5.1.27 Параметр «unbound»

Назначение параметра: настройка unbound.

Таблица 51 — Параметр «unbound»

Параметр	Описание	Тип	Значение по умолчанию
unbound:			
listen_interfaces:			
local_ipv4:			
default:			
docker:			
used:			
interface:			
extra:			
access_control:			
forward_dns:			
local_zone:			
- type:			
zone:			
local_data:			
- domain:			
type:			
ip:			

Примеры корректно настроенных параметров:

```
unbound:
  # Which interfaces will be listened to
  listen_interfaces:
    # interface 127.0.0.1
    local_ipv4: True
```

```
  # interface <ansible_default_ipv4>
  default: True
  docker:
    # If True, role add systemd unit for starting unbound after
    # docker on starting system for binding to docker default interface.
    # Role DO NOT add access_control for docker default net!
    used: True
    # Default: 172.17.0.1
    interface: "172.17.0.1"
  # list of additional interfaces
  extra: []
# The block of addresses with /size allowed to make requests.
access_control:
  - "192.168.103.0/24"
# IP addresses of server to forward for zone ".".
# Default: 8.8.8.8
forward_dns:
  - "8.8.8.8"
  - "8.8.4.4"
#  Configure a local zones.
local_zone:
    # `nodefault` is used to turn off default contents for AS112 zones.
  - type: "nodefault"
    zone: "10.in-addr.arpa"
    # `transparent`:
        # If there is a match from local data, the query is answered.
        # Otherwise if the query has a different name, the query is
        # resolved normally.
  - type: "transparent"
    zone: "10.in-addr.arpa"
    local_data:
      - domain: "example.com"
        type: "A"
        ip: "1.2.3.4"
    # `redirect`:
      # The query is answered from the local data for the zone name.
      # This answers queries for the zone, and all subdomains of the
      # zone with the local data for the zone.
  - type: "redirect"
    zone: "10.in-addr.arpa"
    local_data:
      - domain: "example.com"
```

```
    type: "A"
    ip: "1.2.3.4"
```

5.1.28 Параметр «yum»

Назначение параметра: настройка пакетного менеджера.

Таблица 52 — Параметр «yum»

Параметр **yum:**	**Описание**	**Тип**	**Значение по умолчанию**
configure:	Рекомендуемое значение: False При значении True менеджер обратиться к шаблону настроек		
cron:			
conf:			
installonly_limit:			
set_timeout:			
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
```

6 Предупреждения, выводимые системой

В случае использования неподдерживаемого браузера Система выводит соответствующее уведомление («Ваш браузер ограничивает работу приложения»).

Перечень терминов, определений и сокращений

В таблице представлены сокращения с соответствующими расшифровками, использованные в настоящем документе.

Таблица 53 – Сокращения и расшифровки

Сокращение	Расшифровка
AD	Active Directory (Активный каталог)
CA	Certification authority (Удостоверяющий центр)
CAT	Concatenate (Unix-утилита, последовательно объединяющая указанные файлы)
DNS	Domain Name System (Служба доменных имен)
PKI	Public Key Infrastructure (Инфраструктура открытых ключей)
SSL	Secure Sockets Layer (Криптографический протокол, передающий пакеты данных по защищённым каналам связи)
TCP	Transmission Control Protocol (Протокол управления передачей)
БД	База данных
ДУ	Директория установки
ЕСИА	Единая система идентификации и аутентификации
ОС	Операционная система
ПО	Программное обеспечение
СУБД	Система управления базами данных
ФС	Файловое хранилище