MAILION

Руководство по установке

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

MAILION

2.0

РУКОВОДСТВО ПО УСТАНОВКЕ

Версия 1

На 127 листах

Дата публикации: 12.11.2024

Москва
2024

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

Товарные знаки «МойОфис», «MyOffice», «Squadus», «Mailion» 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.

MAILION

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Сокращения, которые используются в настоящем документе, приведены в таблице 1.

Таблица 1 – Сокращения и расшифровки

Сокращение	Расшифровка
A-запись	Address, одна из ключевых ресурсных записей, используется для связи домена с IP-адресом сервера
DNS	Domain Name System, система доменных имен
FQDN	Fully Qualified Domain Name, полное доменное имя, иногда также называемое абсолютным доменным именем. Это доменное имя, которое указывает точное местоположение домена в древовидной иерархии системы доменных имен (DNS). Включает в себя имена всех родительских доменов иерархии DNS
MX-запись	Mail eXchanger, тип DNS-записи, предназначенный для маршрутизации электронной почты с использованием протокола SMTP
PTR-запись	Pointer, противоположность A-записи для DNS. Связывает IP-адрес сервера с его каноническим именем (доменом). Применяется для фильтрации почты.
SIEM	Security Information and Event Management, управление информацией и событиями безопасности
Standalone	Конфигурация установки ПО «Mailion» без отказоустойчивости
ОС	Операционная система
ПК	Персональный компьютер
ПО	Программное обеспечение
СУБД	Система управления базами данных
УЦ	Удостоверяющий центр
ЦОД	Центр обработки данных

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение

Mailion – корпоративная почтовая система нового поколения на базе микросервисной архитектуры, обеспечивающая обмен электронными сообщениями, планирование рабочего времени, интеллектуальный поиск информации и работу с адресными книгами. Система отличается высокой отказоустойчивостью, способна на быстрое самовосстановление и масштабируемость в зависимости от нагрузок.

В состав продукта входят:

– Почтовая система Mailion для обмена электронными сообщениями, совместной работы с календарями, хранения адресных книг и индексации данных;
– Универсальное приложение Mailion для работы с электронной почтой, календарями, контактными книгами, интеллектуального поиска информации и управления задачами в веб-браузерах и на операционных системах Windows, Linux, macOS;
– Мобильное приложение Mailion для работы с электронной почтой, календарями, контактами и управления задачами с мобильных устройств на операционных системах Android и iOS.

Подробное описание возможностей продукта приведено в документе «Mailion. Функциональные возможности».

1.2 Структура ПО «Mailion»

Структура ПО «Mailion» представляет собой набор сервисов, обеспечивающих работу системы и взаимодействие между подсистемами ПО «Mailion».

Сервисы (представленные в виде установочных ролей) описаны в разделе Описание ролей Ansible.

Общая логическая схема ПО «Mailion» показана на рисунке 1.

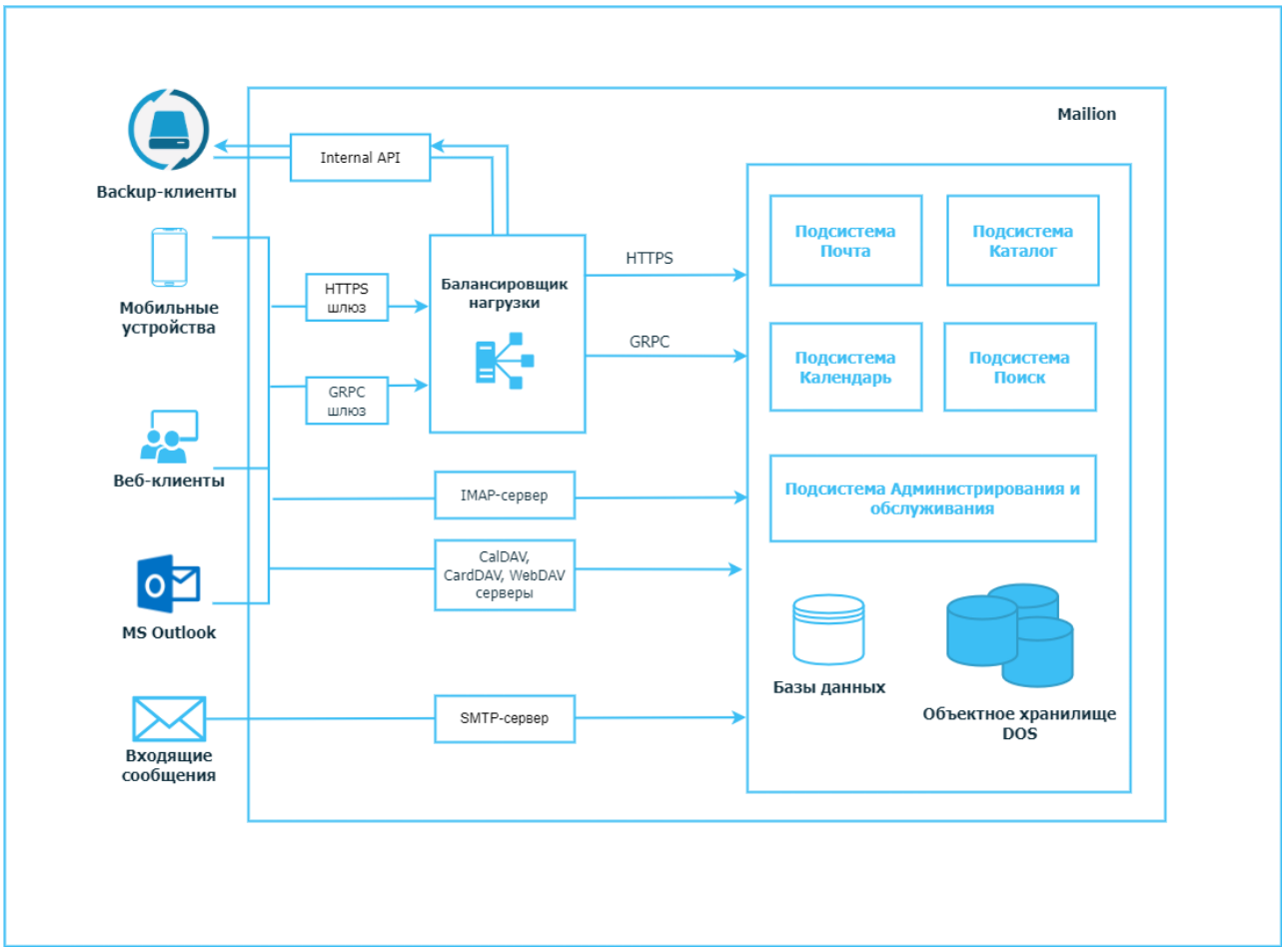


Рисунок 1 – Общая логическая схема ПО «Mailion»

1.3 Требования к квалификации персонала

Администратор ПО «Mailion» должен соответствовать следующим требованиям:

1. Знание основ сетевого администрирования:
   – сетевая модель OSI и стек протоколов TCP/IP;
   – IP-адресация и маски подсети;
   – маршрутизация: статическая и динамическая;
   – протокол обеспечения отказоустойчивости шлюза (VRRP);
2. Опыт работы с подсистемами виртуализации на уровне эксперта:
   – работа с подсистемой контейнерной виртуализации (Docker/Podman);
   – работа с одной из подсистем серверной виртуализации на базе гипервизоров Hyper-V, VMware vSphere ESXi, KVM;

– навык администрирования операционной системы (ОС) Linux с помощью консоли;

– 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;

3. 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);

4. Практический опыт администрирования на уровне эксперта:

– Redis;

– NATS;

– Prometheus;

– MongoDB;

– Postfix.

5. Опыт работы с подсистемой централизованного управления Ansible.
6. Опыт работы со стандартными офисными приложениями.

1.4 Системные требования

Перечень системных требований к аппаратному и программному обеспечению приведен в Аппаратные требования и Программные требования.

1.4.1 Аппаратные требования

Ниже представлены описание ролей групп серверов, стандартные расчеты аппаратной части, расчет на 10 000 пользователей, требования к сетевой и дисковой подсистеме.

1.4.1.1 Описание групп сервера

В таблице 2 приведено обоснование выделения машин под группы сервера.

Таблица 2 – Описание групп сервера

Имя группы сервера	Обоснование выделения машин
ucs_frontend	Веб-серверы Mailion и прокси-сервисы клиентских протоколов. Хранят также статический веб-контент. Должны быть выделенными, так как являются пограничными серверами между внешними сетями и внутренними сервисами системы, или могут быть размещены за пограничным межсетевым экраном для веб-приложений.
ucs_mail	Серверы, выполняющие прием и отправку писем. Определяют границу между внешними сетями и внутренними сервисами. Не рекомендуется совмещать с веб- и прокси-серверами, чтобы при отказе или атаке система не теряла работоспособность в полном объеме. Могут быть размещены за пограничным межсетевым экраном для веб-приложений.
ucs_apps	Серверы основной группы микросервисов, реализующих основной функционал системы
ucs_balancers	
ucs_calendar	
ucs_catalog	Серверы группы микросервисов, реализующих функционал Каталога. Рекомендуется разделение с остальными ролями для обособления в части безопасности. Нагрузка на эту группу повышенная, так как не только пользователи, но и приложения имеют различные уровни доступа, что постоянно проверяется внутри системы
ucs_converter	Серверы группы подготовки предпросмотра документов, конвертации разных форматов в форматы, готовые для отображения в браузере. Отделены от основной функциональности для обеспечения толерантности к отказу, так как работают напрямую с пользовательскими данными, в которых сложно выполнить предпроверку корректности этих данных и отсутствия уязвимостей
ucs_search	Серверы группы подсистемы поиска, обеспечивающей поиск по письмам, вложениям, каталогу, справке. Индексирование данных — ресурсоемкая задача. Чтобы не выделять один мощный сервер, поисковые данные могут быть шардированы для обработки запросов сразу несколькими экземплярами подсистемы поиска

Имя группы сервера	Обоснование выделения машин
ucs_etcd	Серверы группы очередей и хранилищ данных о работе региона. Не имеют тенденции к масштабированию, потому выделены отдельно. Требуют высокоскоростных накопителей, не занятых посторонними задачами
ucs_mq	
ucs_mongodb	Серверы группы баз данных. Требовательны к ресурсам и к гарантиям их наличия
ucs_redis_cache	Серверы группы кеширующих баз данных. Выделены для гарантии обеспечения требуемых ресурсов
ucs_redis_data	
dispersed_object_store	Серверы группы объектного хранилища. Основное хранилище всей системы
ucs_infrastructure	Сервер группы инфраструктуры. Служит для хранения образов инсталляции, сбора журналов доступа и ошибок работы системы, метрик, обеспечивает мониторинг всей системы. Должен быть обособлен для внешнего наблюдения за системой. Его работа не блокирует работу системы

1.4.1.2 Стандартные расчеты аппаратной части

Минимальные требования для установки ПО «Mailion» без отказоустойчивости (Mailion «Standalone») приведены в таблице 3.

⚠ Режим и последующая настройка параметров на оборудовании без поддержки отказоустойчивости предоставляется в целях демонстрации функциональности ПО «Mailion». **Данный режим не поддерживается и к использованию не рекомендуется.**

Таблица 3 – Минимальные требования (установка без отказоустойчивости)

Имя группы сервера	VCPU	RAM, Gb	HDD, Gb (без учета ОС)	SSD, Gb	Количество виртуальных машин	VCPU	RAM, Gb	HDD, Gb (для ОС)	SSD, Gb
	на каждую роль					итого на группу			
Mailion «Standalone»					1	12	32	50	50
ИТОГО:					**1**	**12**	**32**	**50**	**50**

Минимальные требования для установки ПО «Mailion» для отказоустойчивой (кластерной) установки приведены в таблице 4.

Таблица 4 – Минимальные требования (отказоустойчивая установка)

Имя группы сервера	VCPU	RAM, Gb	HDD, Gb (без учета ОС)	SSD, Gb	Количество виртуальных машин	VCPU	RAM, Gb	HDD, Gb	SSD, Gb
	на каждую роль					итого на группу			
ucs_frontend	6	6	10	10	2	12	12	20	20
ucs_mail									
ucs_apps	8	8	10		2	16	16	20	
ucs_catalog									
ucs_calendar									
ucs_balancers									
ucs_converter									
ucs_search	16	18	10	15	3	48	54	30	45
ucs_etcd									
ucs_mongodb									
ucs_redis_cache									
ucs_mq									
ucs_redis_data									
dispersed_object_store	3	4	20	4	4	12	12	80	16
ucs_infrastructure	4	8	100		1	4	8	100	
				ИТОГО:	**12**	**92**	**102**	**250**	**81**

Рекомендованные требования для установки ПО «Mailion» на отказоустойчивом оборудовании приведены в таблице 4.

Таблица 5 – Рекомендованные требования (отказоустойчивая установка)

Имя роли сервера	VCPU	RAM, Gb	HDD, Gb (без учета ОС)	SSD, Gb	Количество виртуальных машин	VCPU	RAM, Gb	HDD, Gb	SSD, Gb
	на каждую роль					итого на группу			
ucs_frontend	4	4	10		2	8	8	20	
ucs_mail	4	4		10	4	16	16		40
ucs_catalog	8	8	10		2	16	16	20	
ucs_apps	8	8	10		2	16	16	20	
ucs_calendar									
ucs_balancers									
ucs_search	4	8		30	3	12	24		90
ucs_converter	4	8		30	3				
ucs_etcd	8	16		30	3	24	48		90
ucs_mongodb									
ucs_mq									
dispersed_object_store	4	4	60	10	4	16	16	240	40
ucs_redis_data	8	8		10	3	24	24		30
ucs_redis_cache									
ucs_infrastructure	4	8	200		1	4	8	200	
ИТОГО:					**26**	**144**	**184**	**510**	**290**

1.4.1.3 Расчёт требований для 10 000 пользователей

Расчет требований для 10 000 пользователей ПО «Mailion» приведен в таблице 6.

Таблица 6 – Расчет требований для 10 000 пользователей

Примечания	Имя группы сервера	VCPU	RAM, Gb	HDD, Gb (без учета ОС)	SSD, Gb	Количество виртуальных машин	VCPU	RAM, Gb	HDD, Gb	SSD, Gb
		на каждую роль					итого на группу			
Веб-сервера Mailion и прокси-сервисы клиентских протоколов, работающие в DMZ	ucs_frontend	**6**	**6**	**50**	**50**	**2**	**12**	**12**	**100**	**100**
ВМ, выполняющие приём и отправку писем, работающие в DMZ	ucs_mail									
Блок ВМ приложений, обеспечивающих основную функциональность	ucs_apps	**8**	**8**	**50**	**0**	**2**	**16**	**16**	**100**	**0**
	ucs_balancers									
	ucs_calendar									
Блок ВМ приложений Каталога	ucs_catalog									
Блок ВМ подсистемы предпросмотра документов	ucs_converter									
Блок ВМ поисковой подсистемы	ucs_search	**8**	**32**	**50**	**256**	**3**	**24**	**96**	**150**	**768**

Примечания	Имя группы сервера	VCPU	RAM, Gb	HDD, Gb (без учета ОС)	SSD, Gb	Количество виртуальных машин	VCPU	RAM, Gb	HDD, Gb	SSD, Gb
		на каждую роль					итого на группу			
ВМ группы очередей и хранилищ данных о работе региона	ucs_etcd	**16**	**32**	**50**	**201**	**3**	**48**	**96**	**150**	**603**
	ucs_mq									
ВМ группы баз данных	ucs_mongodb									
ВМ группы хранилищ	dispersed_object_store	**4**	**8**	**3471**	**24**	**4**	**16**	**32**	**13883**	**95**
ВМ группы кэширующих баз данных	ucs_redis_cache	**8**	**8**	**0**	**50**	**3**	**24**	**24**	**0**	**150**
	ucs_redis_data									
ВМ инфрастуктуры. Является хранилищем всех образов инсталяции, сервером мониторинга, логколлектором	ucs_infrastructure	**4**	**8**	**300**	**0**	**1**	**4**	**8**	**300**	**0**
					ИТОГО:	**18**	**144**	**284**	**14684**	**1716**

Параметры расчета приведены в таблице 7.

Таблица 7 – Параметры расчета

Параметр	Значения	Заполнить	Комментарий
Количество пользователей	10000	да	
Квота на ящик, Гб	1	да	
Избыточность данных DOS: d-сегменты	2	да	Количество сегментов самих данных, при использовании кодов Рида-Соломона, которые будут записаны в хранилище
Избыточность данных DOS: p-сегменты	1	да	Количество избыточных сегментов, при использовании кодов Рида-Соломона, которые будут записаны в хранилище
Фактор репликации данных DOS	3	да	Фактор репликации для индексов DOS (количество полных копий записи индекса)
Фактор репликации данных мета данных	3	да	Фактор репликации для мета данных (заголовки, участники, пр) СУБД Mailion
Процент заполнения квоты ящика	100,00%	нет	
Избыточность данных DOS (d+p)	3	нет	Итоговая избыточность хранилища
Количество писем, шт	20971520	нет	

Дополнительные пояснения приведены в таблице 8.

Таблица 8 – Дополнительные пояснения

Данные, помеченные цветом	Пояснения
	для данных в ячейках, отмеченных этим цветом, нужно 2 или более блочных устройств. Рекомендуются физические устройства, которые не требуют резервирования на уровне RAID массива на хостовой системе
	все ресурсы указаны с расчётом работы ОС ВМ

1.4.1.4 Требования к дисковой подсистеме

Требования к дисковой подсистеме приведены в таблице 9.

Таблица 9 – Характеристики дисков

Тип диска	min IOPS read	min IOPS write	IOPS/GB read	IOPS/GB write	latency (clat) ms
HDD	300	150	1	1	<12
SSD	200000	80000	1700	700	<1

1.4.1.5 Требования к сетевой подсистеме

Между серверами (виртуальными машинами) должен быть канал в 1Гб/с и предельное время ожидания (Network latency) 5-7ms.

1.4.2 Программные требования

Требования к программному обеспечению для места оператора, на котором производится установка, приведены в таблицах 10, 11.

Таблица 10 – Требования к программному обеспечению для места оператора

Требование	Описания	
Поддерживаемые браузеры	Перечень поддерживаемых браузеров приведен в документе «Mailion. Системные требования»	
Python3	v. 3.6+	
Модули Python	jmespath	
	jinja2	необходима версия выше, чем v.2.10 (обновление для CentOS можно выполнить с любого репозитория OpenStack: http://mirror.centos.org/centos/7/cloud/x86_64/openstack-train/ или https://mirror.yandex.ru/centos/7/cloud/x86_64/openstack-train/)
	ansible	2.11 или новее, но до 2.12
	netaddr	python3-netaddr
	dnspython	
	hvac	
	pymongo	Не ниже версии 3.12
Дополнительные пакеты	mongodb-mongosh	Необходима версия 1.6.2_amd64.deb https://www.mongodb.com
	epel-release	Extra Packages Enterprise Linux, https://docs.fedoraproject.org/en-US/epel/

Перед установкой должен быть скачан и смонтирован образ Mailion

Таблица 11 – Требования к программному обеспечению для серверов, на которые производится установка

Требование	Описание
ОС	Перечень поддерживаемых ОС приведен в документе «Mailion. Системные требования»
Стандартные репозитории ОС	Подключение всех стандартных репозиториев ОС либо их зеркал во внутренней сети для установок в закрытом контуре
репозиторий epel (для Centos 7)	Подключение локальной копии репозитория для установок в закрытом контуре
Репозитории elrepo и docker-ce, ppa:canonical-kernel-team/ppa	Подключение репозиториев **elrepo** (http://elrepo.org) и **docker-ce** (https://download.docker.com/linux/centos/docker-ce.repo) для установки соответствующих пакетов ядра Linux и ПО **docker**, не входящих в состав поставки для установок в закрытом контуре
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)
Рекомендации по версии ядра Linux	Требуется ядро mainline (обновляется по умолчанию, если не передан флаг **UPGRADE_KERNEL=false**). С более старыми версиями ядер (lts) работоспособность не гарантируется из-за особенностей **Docker** (требуется полная поддержка cgroup2 в ядре).

1.5 Требования к работе DNS

1.5.1 Организация работы сервисов разрешения имен

Во время установки производится настройка и запуск локального кэширующего DNS-сервера (**unbound**) на серверах группы **ucs_etcd**. Этот сервер используется для запросов только внутри инсталляции и подключается для контейнеров и самих серверов с помощью соответствующих параметров групповых переменных. По умолчанию серверы настроены на работу через **unbound** и не принимают параметры DNS-серверов через **DHCP**. При необходимости **unbound** можно настроить на работу с внутренними DNS-серверами. По умолчанию **unbound** настроен на маршрутизацию запросов на адреса 8.8.8.8 и 1.1.1.1.

1.5.2 Разрешение имен на машине оператора

Перед установкой необходимо убедиться, что на машине оператора доступен и подключен DNS-сервер, в котором созданы записи согласно разделу Внешние DNS-записи.

Должны быть доступны DNS-записи для машин группы **ucs_db**. При необходимости на машине оператора можно отредактировать файл `/etc/hosts` и внести в него соответствующие сопоставления имен и адресов. Пример приведен ниже.

Здесь и далее `<домен_инсталляции>` — это полное доменное имя инсталляции, описанное в разделе Настройка файла hosts.yml.

```
192.168.0.1 ucs-db-1.<домен_инсталляции>
192.168.0.1 mongodb.ucs-db-1.<домен_инсталляции>
.......
192.168.0.n ucs-db-n.<домен_инсталляции>
192.168.0.n mongodb.ucs-db-n.<домен_инсталляции>
```

Проверить разрешение имени машины в адрес можно с помощью команды:

```
> dig A mongodb.ucs-db-1.<домен_инсталляции>
; <<>> DiG 9.18.1-1ubuntu1.2-Ubuntu <<>> A mongodb.ucs-db-1.<домен_инсталляции>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45369
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;mongodb.ucs-db-1.<домен_инсталляции>. IN A

;; ANSWER SECTION:
mongodb.ucs-db-1.<домен_инсталляции>. 900 IN CNAME ucs-db-1.<домен_инсталляции>.
ucs-db-1.<домен_инсталляции>. 900 IN A  192.168.0.1

;; Query time: 23 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Tue Jan 10 15:56:32 MSK 2023
;; MSG SIZE  rcvd: 95
```

Секция `ANSWER SECTION` показывает, что имя разрешается в адрес:

```
mongodb.ucs-db-1.<домен_инсталляции>. 900 IN CNAME ucs-db-1.<домен_инсталляции>.
ucs-db-1.<домен_инсталляции>. 900 IN        A   192.168.0.1
```

1.5.3 Формирование внешних доменных имен инсталляции

При установке системы есть возможность указывать метод формирования доменных имен инсталляции. Шаблон, который формирует итоговый вариант всех DNS-записей, на которых будет работать инсталляция, принимает на вход два параметра:

– **mailion_external_domain** — основной домен, на котором будет работать инсталляция;

– **mailion_domain_module** — шаблон формирования доменного имени.

Примеры работы шаблона показаны в таблице 12.

Таблица 12 — Примеры работы шаблона

mailion_domain_module	Имя сервиса	mailion_external_domain	Результат
`{service}.{domain}`	auth	test.example.com	auth.test.example.com
`{service}-{domain}`	auth	test.example.com	auth-test.example.com
`{service}-xz-1.{domain}`	auth	test.example.com	auth-xz-1.test.example.com

Таким образом, можно гибко настраивать принцип формирования доменных имен инсталляции. Это может пригодиться, например, если имеется Wildcard SSL-сертификат на доменное имя **example.com** и ***.example.com**, но нет на ***.test.example.com**. Можно задать для **mailion_domain_module** значение `{service}-{domain}` и получить домены третьего уровня, которые подходят под текущий Wildcard SSL-сертификат.

1.5.4 Необходимые DNS записи

1.5.4.1 Внешние DNS-записи

В таблицах 13 и 14 приведены все требуемые для инсталляции внешние DNS-записи. Данная таблица сформирована для параметра **mailion_domain_module** со значением `{service}.{domain}` (т. е. для формирования ссылок через точку к указанному домену). Если выбран другой метод формирования, необходимо соотнести его со значениями в таблицах ниже.

Таблица 13 — Сведения о внешних DNS-записях

Имя записи	Тип записи	Значение	Комментарии
api	CNAME	@	
auth	CNAME	@	
autoconfig	CNAME	@	
avatars	CNAME	@	
caldav	CNAME	@	
carddav	CNAME	@	
db	CNAME	@	
@	A	<ucs_frontend_vip>	Значение должно быть равно VIP-адресу серверов группы **ucs_frontend** или IP-адресу единственного сервера этой группы, если производится установка без отказоустойчивости
@	TXT	"v=spf1 mx a:relay.<mailion_external_domain> ~all"	Необходимо указать сформированное имя, с учетом значения в словаре **mailion_external_domain**
@	MX	10 <mx1>	MX-запись указывает на A-запись, в которой содержится адрес первого сервера группы **ucs_mail**
@	MX	10 <mx2>	MX-запись указывает на A-запись, в которой содержится адрес второго сервера группы **ucs_mail** (и т. д.)
grpc	CNAME	@	
imap	CNAME	@	
mail	CNAME	@	
mail._domainkey	TXT	"v=DKIM1; k=rsa; p=<DKIM_KEY>"	Значение **DKIM_KEY** определяется на этапе установки
mx1	A	<ucs_mail_mx[0]>	Внешний IP-адрес, по которому доступен первый сервер группы **ucs_mail**
mx2	A	<ucs_mail_mx[1]>	Внешний IP-адрес, по которому доступен второй сервер группы **ucs_mail** (и т. д.)

Имя записи	Тип записи	Значение	Комментарии
preview	CNAME	@	
relay	A	<ucs_mail_relay_vip>	Значение должно быть равно VIP-адресу серверов группы **ucs_mail** или IP-адресу единственного сервера этой группы, если производится установка без отказоустойчивости
resources	CNAME	@	
secured	CNAME	@	
smtp	A	<ucs_mail_vip>	
_adsp._domainkey	TXT	"dkim=all"	

Таблица 14 – Сведения о внешних DNS-записях

Имя записи	Тип	Приоритет	Вес	Порт	Адрес
_autodiscover._tcp	SRV	0	0	443	<mailion_external_domain>.
_caldavs._tcp	SRV	0	0	6787	caldav.<mailion_external_domain>.
_carddavs._tcp	SRV	0	0	6787	carddav.<mailion_external_domain>.
_grpcsec._tcp	SRV	0	0	3142	grpc.<mailion_external_domain>.
_imap._tcp	SRV	0	0	143	imap.<mailion_external_domain>.
_imaps._tcp	SRV	10	0	993	imap.<mailion_external_domain>.
_smtps._tcp	SRV	10	0	465	smtp.<mailion_external_domain>.
_submission._tcp	SRV	0	0	587	smtp.<mailion_external_domain>.
_submissions._tcp	SRV	10	0	465	smtp.<mailion_external_domain>.

Примеры написания DNS-записей приведены в приложении Пример написания внешних DNS-записей.

1.5.4.2 Внутренние DNS-записи

Все DNS-записи, используемые для работы самой системы внутри контура установки, формируются через точку относительно вписанного в файл **inventory** имени сервера и создаются в **unbound** автоматически на основе переменной **ansible_default_ipv4**.

Это поведение можно переопределить, если заполнить все адреса вручную на основе примеров в файле групповых переменных или если не использовать **Ansible** и заполнить все необходимые записи во внешнем DNS-сервере. При подобном варианте необходимо создать A-записи для каждого сервера, вписанного в файл **inventory**, а также CNAME адреса на все поддомены (*) к каждому серверу, вписанному в файл **inventory**.

Пример заполнения таких записей приведен в таблице 15.

Таблица 15 – Пример заполнения

Имя записи	Тип записи	Значение
infra-01	A	10.10.1.110
*.infra-01	CNAME	infra-01

unbound не должен быть доступен из внешней сети.

Использование **unbound** необязательно. Если при заполнении файла с параметрами групповых переменных задается параметр **mailion_use_unbound: False**, то **unbound** будет установлен, но не будет принимать участия в работе ПО «Mailion».

1.6 Рекомендации

1.6.1 Рекомендации по использованию файловых систем

В соответствии с рекомендациями производителей операционных систем для ОС CentOS следует использовать файловую систему XFS.

1.6.2 Рекомендации по разметке дисков

При разметке дисков требуется учитывать следующее:

- все рекомендуемые аппаратные требования приведены в разделе Аппаратные требования, в соответствии с приведенными в разделе таблицами для разных типов установки будут разные требования по выделяемому дисковому пространству;
- для всех серверов рекомендуется оставлять не менее 20 Гб на корневой раздел для штатной работы ОС.
- для роли **ucs_infrastructure** или инсталляции в режиме «Standalone» рекомендуется выделить 50 Гб на корневой раздел, так как во время установки все образы инсталляции предварительно копируются в локальное хранилище docker (`/var/lib/docker/`);

- для всех серверов рекомендуется выделять отдельный раздел /srv, в который происходит установка компонентов системы, и переполнение которого не приведет к аварийной работе самой ОС. В этот раздел также могут быть направлены копии журналов работы компонентов, при соответствующей настройке лог-коллектора, что потребует дополнительного дискового пространства;
- для сервера роли **dispersed_object_store** рекомендуется выделять независимые диски HDD для серверной части и диски SSD под метаданные. Например:
  - **/srv/docker/dispersed_object_store/data/metadata/** – SSD, индексы документов и сегментов;
  - **/srv/docker/dispersed_object_store/data/disk1/{blob,rocksdb}** – HDD1, бэкенд1 – блоб и индекс бэкенда;
  - **/srv/docker/dispersed_object_store/data/disk2/{blob,rocksdb}** – HDD2, бэкенд2 – блоб и индекс бэкенда;
- распределение сегментов (data segments) + (parity segments):
  - сумма data segments + parity segments не должна превышать количества независимых дисков в серверной части хранилища;
  - не менее 2 + 1 независимых дисков в серверной части хранилища;
  - для кластера из трех машин минимально допустимые значения – 2 (data segments) + 1 (parity segments) сегментов.

1.7 Ограничения

1.7.1 Ограничение на количество администраторов тенанта

При развертывании инсталляции можно создать только одного администратора тенанта. После завершения развертывания можно добавить произвольное количество администраторов тенанта.

1.7.2 Ограничения при выполнении кластерной установки

При кластерной установке ПО «Mailion» можно выделить отдельный сервер для каждой роли или совместить несколько ролей на одном сервере. Необходимо учитывать, что некоторые серверные роли могут быть не совместимы с другими ролями.

Пример совместимости ролей приведен в таблице 16.

Таблица 16 – Совместимости ролей

Имя роли сервера	Совместимость с другими ролями сервера
ucs_calendar	Совместимы с другими ролями
ucs_balancers	
ucs_mq	

ucs_mail	Несовместимы с ролями **ucs_mongodb**, **ucs_etcd**, **ucs_redis_cache**, **ucs_redis_data**
ucs_apps	
ucs_catalog	
ucs_converter	
ucs_etcd	Несовместимы с ролями **ucs_apps**, **ucs_mail**, **ucs_converter**, **ucs_catalog**
ucs_mongodb	
ucs_redis_cache	
ucs_redis_data	
ucs_frontend	Несовместимы с другими ролями
ucs_search	
dispersed_object_store	
ucs_infrastructure	

Не рекомендуется совмещать серверные роли при установке

1.7.3 Ограничение по работе с файлом inventory

В файл `hosts.yml` вносятся только полностью определенные имена доменов (FQDN). Эти имена будут использоваться во время и после установки системы для обращения к внутренним сервисам. Часть логики установщика использует их для формирования доменных имен и адресов сервисов.

1.7.4 Ограничение по работе с Ansible

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

Важно самостоятельно установить необходимые модули python из раздела Программные требования, так как они не являются частью поставки системы.

1.7.5 Ограничение по работе с системами виртуализации

Следующие системы виртуализации поддерживаются для обеспечения работы ПО «Mailion»:

- VMware;
- KVM.

1.7.6 Ограничение по работе с хостами MX

Каждый хост MX должен иметь PTR-запись для обеспечения правильной фильтрации писем антиспам-системой.

1.7.7 Ограничение при заполнении файлов переменных

При заполнении инвентарного файла имя **tier** (#SECTION 2) должно всегда начинаться с «ucs_».

1.7.8 Ограничение при использовании данных внешнего каталога

Необходимо использовать учетные данные внешнего LDAP-каталога для авторизации и отправки писем в ПО «Mailion». Если пользователь хочет отправить письмо на адрес **test@installation.net**, то письмо не отправится, так как на домене **installation.net** нет почтового сервиса. Поэтому необходимо заменить доменную часть в Email при отправке письма.

Например, в ПО «Mailion» создан домен **ipa.example.installation.net**, на нем есть почтовый сервис и он связан с **example.ru** через поле **x_external_names** в базе данных. Соответственно, отправить письмо необходимо на адрес **test@ipa.example.installation.net** .

Важно. Если этот пользователь еще не был создан в ПО «Mailion» (а при отправке письма на почту из внешнего каталога в ПО «Mailion» создается пользователь, если он еще не был синхронизирован), то для того, чтобы была возможность в будущем под ним авторизоваться, необходимо использовать для входа не адрес **test@ipa.example.installation.net**, на который осуществлялась отправка письма, а **test@installation.net** по причине того, что такой Email заведен во внешнем каталоге.

1.8 Типовые схемы установки

ПО «Mailion» может быть представлено следующими типами установки:

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

2 ПЕРВИЧНАЯ УСТАНОВКА

2.1 Дистрибутив

Дистрибутив Mailion поставляется в виде файла образа ISO с именем `Mailion_[RELEASE].iso`.

После копирования инсталляционного архива необходимо проверить его контрольную сумму MD5 и SHA256, сверив ее с полученной от вендора ПО.

Образ дистрибутива предварительно монтируется командой:

```
mount Mailion_[RELEASE].iso /mnt/disk
```

В состав дистрибутива ПО «Mailion» входят:

1. Установщик рабочего места оператора (`mailion_ansible_bin_[RELEASE].run`).
2. Установщик окружения для проведения установки, включающий все необходимые образы и пакеты (`mailion_infra_[RELEASE].run`).
3. Файлы EULA (End-user license agreement).
4. Файлы TPL (Third-party license).

Где `[RELEASE]` — обозначение текущего релиза.

2.2 Подготовка к установке

В данном разделе приведена последовательность действий, которую необходимо выполнить перед установкой Mailion.

2.2.1 Описание ролей Ansible

Ansible применяется для автоматизации настройки и развертывания сервисов. Список ролей Ansible для преднастройки серверов перед установкой ПО «Mailion» приведен в таблице 17.

Таблица 17 – Роли Ansible для преднастройки серверов перед установкой

Наименование роли	Описание
authorized_keys	Добавляет указанные SSH-ключи для выбранных пользователей на серверы группы **play_hosts**
hostname	Устанавливает **hostname** для выбранных серверов
SELinux	Проверяет режим работы **SELinux** и переключает его в режим «enforcing» (только для дистрибутивов с пакетным менеджером **yum**)
packagemanager	Настраивает пакетный менеджер
locale	Устанавливает параметры **locale** на серверах
timezone	Устанавливает часовой пояс на серверах

Наименование роли	Описание
sshd	Производит настройку службы удаленного доступа **sshd**
chrony	Устанавливает и настраивает службу синхронизации времени **chronyd** (только для ОС на базе Red Hat)
timesyncd	Устанавливает и настраивает службу синхронизации времени **timesyncd** (только для ОС Astra Linux)
sysctl	Устанавливает требуемые параметры ядра на серверах
limits	Настраивает параметры ограничений на серверах
kernel_ml	Устанавливает пакет **kernel_ml** последнего доступного ядра
kernel_ml_deb	Устанавливает пакет **kernel_ml** последнего доступного ядра для ubuntu
rsyslog	Устанавливает и настраивает сервис сбора журналов
docker	Устанавливает и настраивает **Docker**, подключает к **docker registry**
unbound	Устанавливает и настраивает кэширующий DNS-сервер
iptables	Устанавливает и настраивает службы межсетевого экрана с параметрами, требуемыми для конкретной роли
resolv	Производит настройку файла **resolv.conf**
package_tools	Добавляет требуемые пакеты для работы ПО «Mailion» в целевую ОС

Роли, используемые для подготовки ПО «Mailion» описаны далее в таблице 18.

Таблица 18 – Описание ролей, используемых при подготовке ПО «Mailion»

Наименование роли	Описание
keepalived	Устанавливает и запускает службу, реализующую протокол VRRP
cAdvisor	Устанавливает сервис **cAdvisor**, осуществляющий сбор метрик работы контейнеров
node_exporter	Устанавливает сервис **node_exporter**, осуществляющий сбор метрик работы сервера
node_cert_exporter	Мониторинг срока действия сертификатов
node_filestat_exporter	Мониторинг появления дампов памяти
blackbox_exporter	Мониторинг доступности веб-интерфейса
syslog_ng	Устанавливает сервис централизованного сбора журналов работы системы
logrotate	Настраивает ротацию хранимых журналов работы системы
ca	Устанавливает и настраивает сервис внутреннего центра сертификации
alertmanager	Устанавливает и настраивает сервис оповещений о событиях мониторинга
devkalion	Устанавливает и настраивает сервис автообнаружения сервисов инсталляции для мониторинга
gesiona	Устанавливает и настраивает сервис, экспортирующий список сервисов инсталляции для сервиса мониторинга
prometheus	Устанавливает и настраивает сервис мониторинга
grafana	Устанавливает и настраивает сервис отображения данных мониторинга инсталляции

Наименование роли	Описание
kunkka	Устанавливает и настраивает сервис отображения данных о запущенных контейнерах на каждом сервере и их конфигурационных файлов
plugin_certificate	Роль, выписывающая сертификат для сборки клиентских приложений **outlook plugin**
etcd	Устанавливает базу данных **etcd**
hydra	Устанавливает и настраивает сервис обнаружения и балансировки нагрузки gRPC
nats	Устанавливает и настраивает **NATS**
nats_exporter	Сбор метрик мониторинга с **NATS**
mongodb	Устанавливает и настраивает документоориентированную СУБД
mongodb.mailion_migration	Устанавливает миграции данных сервисов в базах **MongoDB**
mongodb_exporter	Сбор метрик мониторинга с **MongoDB**
dorofej	Роль работы с модулем **Ansible**, реализующим первичную миграцию СУБД
redis	Устанавливает и настраивает кластер хранилищ **Redis**
theseus	Устанавливает и настраивает сервис работы с учетными данными
perseus	Устанавливает и настраивает сервис хранения контактов
erakles	Устанавливает и настраивает сервис работы с сущностями
odusseus	Устанавливает и настраивает сервис работы с регионами
talaos	Устанавливает и настраивает сервис работы с тенантами
daidal	Устанавливает и настраивает сервис работы с доменами
minos	Устанавливает и настраивает сервис работы с сессиями
ektor	Устанавливает и настраивает сервис работы со связями, сущностями
pasifae	Устанавливает и настраивает сервис подсказок при поиске
dispersed_object_store	Устанавливает и настраивает объектное хранилище, предоставляющее gRPC-интерфейс для хранения бинарных данных и метаданных
achill	Устанавливает и настраивает сервис работы с аватарками
jod	Устанавливает и настраивает сервис для конвертации документов
pregen	Устанавливает и настраивает сервис для конвертации документов
cvm	Устанавливает и настраивает сервис для конвертации документов
cu	Устанавливает и настраивает сервис для конвертации документов
sdd	Устанавливает и настраивает сервис для конвертации документов
meepo	Устанавливает и настраивает сервис генерации превью
mailbek	Устанавливает и настраивает сервис проксирования запросов к шардированным данным на экземплярах поисковой системы
dirbek	Сервис поиска по каталогу
helpbek	Устанавливает и настраивает поисковый сервис по имеющейся веб-документации инсталляции
tripoli	Устанавливает и настраивает единый индексно-поисковый сервис
rspamd	Устанавливает и настраивает сервис антиспама

Наименование роли	Описание
zeus	Устанавливает и настраивает сервис, отвечающий за шаблонизацию и настройку работы с письмами
paranoid	Устанавливает и настраивает сервис, реализующий протоколы **Postfix Policy Delegation** и **Nginx HTTP Auth**
woof	Устанавливает и настраивает сервис, реализующий метод **search** протокола LDAP для резолвинга групповых адресов, алиасов, получения списка доменов со стороны **postfix**
ariadne	Сервис аутентификации для MTA
lmtp	Устанавливает и настраивает сервис, реализующий протокол **lmtp**
postfix	Устанавливает роль для развертывания почтового сервера (**MTA**)
nginx	Устанавливает и настраивает сервер nginx в режиме **smtp**
kongur	Устанавливает и настраивает сервис, отвечающий за работу календарных событий
mars	Сервис для взаимодействия со ПО Squadus (создание и редактирование чатов и конференций)
kex	Устанавливает и настраивает сервис проксирования запросов к внешним календарям
thoth	Устанавливает и настраивает сервис сохранения полей
ares	Устанавливает и настраивает сервис для взаимодействия с системами видеоконференций
othrys	Устанавливает и настраивает взаимодействия с внешними календарными серверами
elysion	Устанавливает и настраивает сервис выполнения асинхронных работ в календаре
mosquito	Устанавливает и настраивает сервис, предоставляющий абстракцию pub/sub над **AMQP**
viper	Устанавливает и настраивает сервис для сохранения писем в системе
razor	Устанавливает и настраивает сервис для отправки писем по шаблону с локализацией
weaver	Устанавливает и настраивает сервис для построения всего сообщения (его web-представления) или его части (для **IMAP**)
marker	Устанавливает и настраивает сервис для управления тегами
hog	Устанавливает и настраивает сервис для получения и сохранения настроек пользователей
beef	Устанавливает и настраивает сервис для сохранения и получения метаданных писем
mixer	Устанавливает и настраивает сервис для получения объектов веб-интерфейсом
atlas	Устанавливает и настраивает сервис для отправки почтовых сообщений
kronos	Устанавливает и настраивает сервис, предназначенный для регистрации задач на отложенное исполнение операций
clotho	Устанавливает и настраивает сервис для хранения истории изменений объектов и тегов
orpheus	Устанавливает и настраивает сервис проксирования аутентификации и поиска сущностей
iason	Устанавливает и настраивает сервис контроля за регистрацией внешних пользователей
cleanup	Производит полное удаление выбранных компонентов (при необходимости)

Наименование роли	Описание
imap	Устанавливает и настраивает сервис, реализующий протокол **IMAP**
cox	Устанавливает и настраивает **proxy grpc** сервис
house	Устанавливает и настраивает веб-сервер
ararat	Устанавливает и настраивает сервис для работы настольных и мобильных клиентов с календарем по протоколу CalDAV/CardDAV
leda	Устанавливает и настраивает LDAP прокси-сервер
sophokles	Устанавливает и настраивает сервис авторизации
dafnis	Устанавливает и настраивает сервис квот
iolaos	Устанавливает и настраивает сервис создания динамических групп
homeros	Устанавливает и настраивает сервис аудита действий пользователя.
adonis	Устанавливает и настраивает сервис для административных функций ministerium
etcd.etcd_backup	Настройка автоматического резервного копирования для **etcd**
mongodb.mongodb_backup	Настройка автоматического резервного копирования для **MongoDB**
sreindexer	Настройка инструмента для переиндексации поиска
nats.nats_backup	Настройка автоматического резервного копирования **NATS**
themis	Устанавливает и настраивает сервис для генерации ссылок или занятость пользователей

2.2.2 Подготовка инфраструктуры установки

Для подготовки инфраструктуры установки должны быть проведены следующие действия (последовательность не важна):

- Установка хранилища образов **Docker** (**docker_registry**) на машине **ucs-infrastructure**, см. раздел Установка хранилища образов Docker.
- Установка подсистемы управления конфигурациями (**Ansible**) на машине оператора, см. раздел Установка конфигурационных файлов Ansible.

2.2.2.1 Установка хранилища образов Docker (docker_registry)

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

Для установки необходимо:

1. Скопировать файл `mailion_infra_[RELEASE].run` в домашний директорий пользователя.
2. Запустить скрипт установки:

```
bash mailion_infra_[RELEASE].run
```

3. Дождаться проверки целостности файла и его распаковки.

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

4. Согласиться на продолжение установки, нажать «**Y**».

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

5. Указать тип контейнерной виртуализации (**docker** или **podman**, см. варианты установки в разделе Запуск установки).

```
choose container_management_tool ('docker' or 'podman')*:
```

6. Во время установки на экране пользователя будет отображен список выполненяемых операций и их статус:

```
.....................................................          ........
Check if container with registry is available                  [  OK  ]
Ensure that registry configuration directory exists            [CHANGE]
Ensure that docker-registry env file exists                    [CHANGE]
Check if old registry data directory exists                    [  OK  ]
Ensure that registry data directory exists                     [CHANGE]
Ensure that container with registry is available               [CHANGE]
Ensure that docker-registry is running                         [  OK  ]
Extracting registry archive...                                 [  OK  ]
Remove dangling and outdated images                            [  OK  ]
.....................................................          ........
```

Необходимо убедиться, что элементы списка содержат статус `[ OK ]` или `[CHANGE]`, это свидетельствует об успешной установке компонента.

При получении статуса `[FAIL]` для любого из компонентов необходимо обратиться в техническую поддержку.

Установка хранилища образов **Docker** (**docker_registry**) будет считаться успешно завершенной в случае успешной установки всех компонентов.

2.2.2.2 Установка конфигурационных файлов Ansible для развертывания ПО «Mailion»

Установка производится на рабочем месте оператора. Перед началом установки необходимо проверить следующие условия:

- вход выполнен под пользователем **root** или под пользователем **sudo** с привилегиями **yum** (**dnf7**);
- машина, на которой выполняется установка, соответствует требованиям, приведенным в разделе Системные требования;
- с выбранного сервера есть возможность доступа по SSH к другим серверам, на которых выполняется установка;
- система управления конфигурациями **Ansible** установлена, другие конфигурационные файлы **Ansible** в системе отсутствуют;
- необходимые модули установлены в системе, их версии соответствуют требованиям.

В системе управления конфигурациями не должно быть предыдущих конфигурационных файлов самой системы. 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 разделе https://docs.ansible.com/ansible/latest/reference_appendices/config.html#the-configuration-file

Перед установкой важно самостоятельно установить необходимые модули python из раздела Программные требования, так как они не являются частью поставки системы.

Для установки необходимо:

1. Скопировать файл `mailion_ansible_bin_[RELEASE].run` в домашнюю директорию пользователя.
2. Запустить скрипт установки:

```
bash mailion_ansible_bin_[RELEASE].run
```

3. Согласиться на продолжение установки, нажать на клавишу «**Y**».

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

4. Во время установки на экране пользователя отображается список выполняемых операций и их статус:

```
.............................                                    ........
Create playbooks symlink                                         [  OK  ]

Create group_vars directory                                      [  OK  ]
```

```
Create group_vars/all symlink                                    [  OK  ]
Create host_vars directory                                       [  OK  ]
Create certificates directory                                    [  OK  ]
Create certificates symlink                                      [  OK  ]
.............................                                    ........
```

Необходимо убедиться, что все операции имеют статус `[ OK ]`. При появлении статуса `[FAIL]` для любого из компонентов необходимо обратиться в техническую поддержку. Установка конфигурационных файлов **Ansible** считается выполненной в случае успешной установки всех компонентов.

2.2.2.3 Установка ПО «Mailion» с машины оператора

К началу данного этапа структура каталогов инсталляции `~/install_mailion/` должна выглядеть следующим образом (рис. 2):

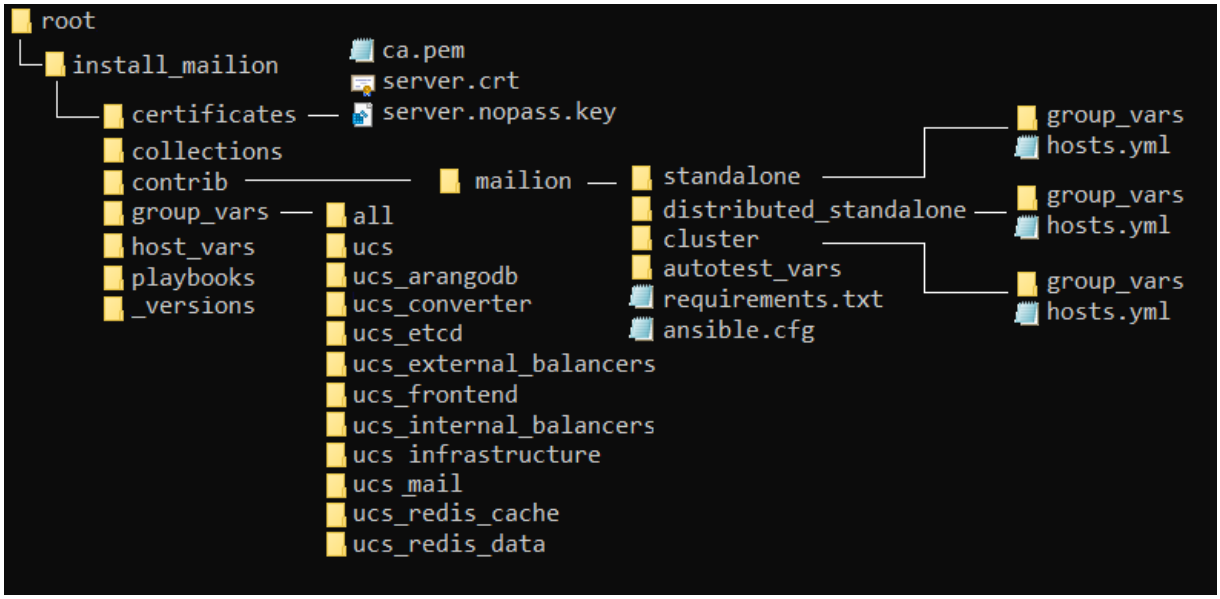


Рисунок 2 – Структура каталогов перед началом инсталляции

В инсталляторе представлены предзаполненные файлы конфигураций (установка описана в разделе Установка конфигурационных файлов Ansible), которые помогут в настройке необходимого функционала будущей системы. В каталоге `~/install_mailion/contrib/mailion/` находятся три подкаталога, соответствующие возможным конфигурациям установки:

- `cluster` (кластерная конфигурация);
- `standalone` («Standalone»);
- `distributed_standalone` (распределенная конфигурация «Standalone»).

Так как целевое назначение системы – крупная отказоустойчивая инсталляция, в данном документе будет описана **кластерная** конфигурация установки.

При установке конфигурации «**Standalone**» необходимо воспроизвести аналогичные этапы установки, описанные в данном разделе. Отличие будет заключаться в названии каталога, в котором находится конфигурационный файл для данной конфигурации.

Перед установкой необходимо перейти в каталог `~/install_mailion/` с помощью команды:

```
cd ~/install_mailion
```

Данный каталог будет являться корневой точкой установки

2.2.2.3.1 Копирование файла ansible.cfg

Необходимо скопировать конфигурационный файл `ansible.cfg` из каталога `~/install_mailion/contrib/mailion/ansible.cfg` в корневой каталог установки с помощью команды:

```
cp contrib/mailion/ansible.cfg .
```

2.2.2.3.2 Настройка файла hosts.yml

Для настройки файла **inventory** (`hosts.yml`) необходимо:

1. Предварительно скопировать файл из каталога с заполненными шаблонами `~/install_mailion/contrib/mailion/<конфигурация_установки>`. Для кластерной конфигурации следует воспользоваться следующей командой:

```
cp contrib/mailion/cluster/hosts.yml .
```

Для конфигурации «Standalone» необходимо использовать файл hosts.yml из каталога `standalone`.

Для распределенной конфигурации «Standalone» необходимо использовать файл hosts.yml из каталога `distributed_standalone`.

2. Открыть файл `hosts.yml` в редакторе и заменить все текстовые вхождения «`installation.example.net`» на доменное имя инсталляции (имя указывается в нижнем регистре). Важно не менять имена хостов до `.installation.example.net`, можно менять только их количество.

В файл `hosts.yml` вносятся только полные доменные имена (FQDN). Эти имена будут использоваться во время и после установки системы для обращения к внутренним сервисам. Часть логики установщика использует их для формирования доменных имен и адресов сервисов.

3. Если на машине оператора планируется использовать несколько инсталляций Mailion, то необходимо заменить в файле `hosts.yml` имя группы **ucs_setup** (в секции `## SECTION 2`) на имя текущей инсталляции (префикс **ucs_** следует оставить в имени, например: **ucs_mailion**). Аналогичным образом нужно поменять значение переменной **tier** (рис. 3).

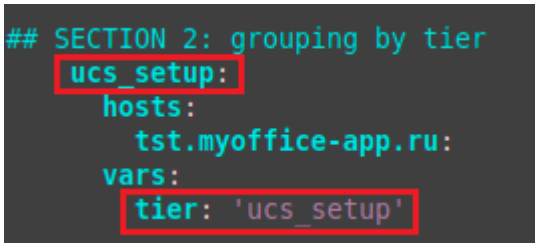


Рисунок 3 — Настройка имени инсталляции

2.2.2.3.3 Копирование каталога групповых переменных

Для подготовки каталога групповых переменных необходимо:

1. Создать в каталоге групповых переменных (`group_vars`) подкаталог для серверов с именем `<имя_инсталляции>` группы инсталляции из файла `hosts.yml`. Имя данного подкаталога должно совпадать с именем инсталляции из секции `## SECTION 2` (по умолчанию — **ucs_setup**, либо другое заданное имя).

```
cd group_vars
mkdir <имя_инсталляции>
```

2. Для **кластерной** установки скопировать в каталог групповых переменных (`group_vars`) каталог с переменными для заполнения:

```
cp -r contrib/mailion/cluster/group_vars/ucs_setup/*
group_vars/<имя_инсталляции>
```

Для установки **standalone** необходимо скопировать конфигурационный файл из папки `contrib/mailion/standalone`.

```
        cp -r contrib/mailion/standalone/group_vars/ucs_setup/*
group_vars/<имя_инсталляции>
```

2.2.2.3.4 Настройка файла main.yml

Открыть файл `main.yml` из каталога `group_vars/<имя_инсталляции>` (см. предыдущий раздел) и отредактировать значения параметров, которые находятся в комментариях. Набор обязательных для настройки параметров приведен в разделе Минимальный набор параметров.

В случае если данная инсталляция Mailion будет использоваться для восстановления из резервной копии предыдущей инсталляции, то необходимо задать идентификатор региона данной инсталляции в переменной `dorofej_region_id`.

При необходимости хранения паролей в зашифрованном виде следует зашифровать содержимое файла `main.yml` с помощью команды:

```
    ansible-vault encrypt group_vars/ucs_setup/main.yml --ask-vault-pass
```

Затем ввести пароль для шифрования. Для удобства можно использовать файл с парольной фразой. Для этого необходимо создать текстовый файл с паролем. В таком случае команда будет следующей:

```
    ansible-vault encrypt group_vars/ucs_setup/main.yml --vault-password-
file=.filesecret
```

Чтобы отменить шифрование файла необходимо опцию **encrypt** в команде заменить на **decrypt**. Чтобы отредактировать зашифрованный файл, следует выполнить команду:

```
    ansible-vault edit group_vars/ucs_setup/main.yml --vault-password-
file=.filesecret* (или --ask-vault-pass)
```

2.2.2.3.5 Настройка файла ministerium.yml

Открыть файл `ministerium.yml` и задать значения параметров, следуя указаниям в комментариях. Примеры задания параметров можно найти в разделе Настройка основных параметров установки.

2.2.3 Установка и обновление пакетов Python

Требуется наличие программного обеспечения, описанного в разделе Программные требования, для чего необходимо на машине оператора установить или обновить следующие пакеты:

– Установка или обновление каталога пакетов python:

```
python3 -m pip install --upgrade pip==20.3.4
```

– Установка модуля ansible-core (версия может отличаться):

```
pip3 install --no-cache-dir hvac ansible-core==2.11.9
```

– Установка необходимых зависимостей:

```
pip3 install --no-cache-dir -r
~/install_mailion/contrib/mailion/requirements.txt
```

2.2.4 Размещение ssl-сертификатов для шифрования

Имена сертификатов могут быть произвольными, но они потребуются для дальнейшего заполнения параметров групповых переменных, поэтому важно их запомнить. В файле групповых переменных `extra_vars.yml`, создание которого было описано в разделе Копирование папки групповых переменных, заполнены имена сертификатов по умолчанию. Если назвать файлы сертификатов соответствующим образом, то менять имена в переменных не нужно.

Состав необходимых сертификатов:

1. Сертификат внешнего домена `server.crt`.
2. Ключ внешнего домена `server.nopass.key`.
3. Цепочка сертификатов промежуточных центров сертификации (CA) внешнего домена `ca.pem`.

Формат файла: в конце файла не должно быть пустой строки.

```
cat certificates/server.crt

-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

Необходимо скопировать файлы сертификатов (`ca.pem`, `server.crt`, `server.nopass.key`) в папку:

```
~/install_mailion/certificates/
```

Имена ключей групповых переменных находятся в переменных **mailion_external_cert_filename**, **mailion_external_key_filename**, **mailion_external_ca_filename**:

```
mailion_external_cert_filename: "server.crt"
mailion_external_key_filename: "server.nopass.key"
mailion_external_ca_filename: "ca.pem"
```

При установке ПО «Mailion» есть возможность использования сертификатов центра Let's Encrypt на усмотрение администратора установки.
Разработчик ПО «Mailion» не несет ответственности за получение, обновление и управление сертификатами Let's Encrypt

В случае использования самоподписанного сертификата в конфигурационный файл необходимо добавить флаг: **mailion_use_self_signed_external_certificate: true**

2.2.5 Настройка основных параметров установки

2.2.5.1 Минимальный набор параметров

Минимальный набор обязательных параметров:

- ansible_user;
- codec_secret_key;
- dispersed_object_store_management_token;
- grafana_admin_password;
- house_ldapauth_password_salt;
- hydra_get_service_list_token;
- jwt_key;
- keepalived_vrrp_instances;
- mailion_cluster, mailion_domain_module, mailion_external_domain, mailion_installation_admin_password, mailion_integrations, mailion_internal_web_auth, mailion_max_users, mailion_service_accounts, mailion_supported_domains, mailion_tenants;
- mongodb_root_password, mongodb_secured_key, mongodb_management_users;
- nats_authorization_password, nats_cluster_authorization_password;
- redis_cluster_replicas, redis_dafnis_password, redis_dowal_password, redis_ektor_password, redis_erakles_password, redis_euripides_password, redis_hog_password, redis_homeros_password, redis_leda_password, redis_minos_password, redis_rspamd_password, redis_sdd_password, redis_viper_password;
- rspamd_kse_endpoints, rspamd_dkim_hosts, rspamd_web_password;
- servus;
- sophokles_access_token;

- theseus_cipher_key;
- tls_certs_remote_token_key;
- unbound_forward_addresses.

Структура и способы задания указанных параметров приведены в разделах ниже.

2.2.5.1.1 Настройка параметров установки ansible_user

Настройка параметров приведена в таблице 19.

Таблица 19 – Настройка параметров ansible_user

Параметр	Тип данных	Описание
ansible_user:	str	Имя пользователя, под которым установщику будут доступны серверы инсталляции по **ssh**

Пример корректно настроенного параметра:

```
ansible_user: "root"
```

2.2.5.1.2 Настройка параметров codec_secret_key

Настройка параметров приведена в таблице 20.

Таблица 20 – Настройка параметров codec_secret_key

Параметр	Тип	Описание
codec_secret_key:		Словарь параметров секретов для формирования шифрованной ссылки
rcr:	str	Используется для формирования ссылки на проксирование данных внутри системы
secret_link:	str	Используется для формирования ссылки на проксируемые ресурсы
values_codec	str	Значение

Пример корректно настроенного параметра:

```
codec_secret_key:
  rcr: "O1Wk7ha80M1qfvq8UtuZg918AZyh+q65s68dKvXwVTQ="
  secret_link: "69rUgWgrLbV50CiAEK78AJIrLoWBGHGwYCX25phh3yg="
  values_codec: "ggxhfxrjshb034fosedfwd3d"
```

2.2.5.1.3 Настройка параметров dispersed_object_store

Настройка параметров приведена в таблице 21.

Таблица 21 – Настройка параметров dispersed_object_store

Параметр	Тип	Описание
dispersed_object_store_management_token: ""	str	Токен доступа для управления через API сервиса

Пример корректно настроенного параметра:

```
dispersed_object_store_management_token: "Aig2utoavi6iageiltas"
```

2.2.5.1.4 Настройка параметра Docker

Настройка параметров приведена в таблице 22.

Таблица 22 – Настройка параметров Docker

Параметр	Тип	Описание
docker_daemon_parameters:		Параметры демона **docker**
bip:	str	Подсеть и маска для **docker**
dns:	list	Список строк с адресами DNS-серверов
mtu:	int	Значение **MTU** для сетевого интерфейса **docker**

Пример корректно настроенного параметра:

```
docker_daemon_parameters:
  bip: "172.17.0.1/16"
  dns:
      - "8.8.8.8"
      - "1.1.1.1"
  mtu: 1412
```

2.2.5.1.5 Настройка параметров grafana

Настройка параметров приведена в таблице 23.

Таблица 23 – Настройка параметров grafana

Параметр	Тип	Описание
grafana_admin_password:	str	Пароль администратора grafana

Пример корректно настроенного параметра:

```
grafana_admin_password: "Ooj0Inahgh2Ixailoxie"
```

2.2.5.1.6 Настройка house

Настройка параметров приведена в таблице 24.

Таблица 24 – Настройка параметров house

Параметр	Тип	Описание
house_ldapauth_password_salt:	str	Соль для хеширования паролей при LDAP-авторизации

Пример корректно настроенного параметра:

```
house_ldapauth_password_salt: ")6_]*|,)(bJ;PN"
```

2.2.5.1.7 Настройка hydra

Настройка параметров приведена в таблице 25.

Таблица 25 – Настройка параметров hydra

Параметр	Тип	Описание
hydra_get_service_list_token:	str	Токен для обращения в API сервиса

Пример корректно настроенного параметра:

```
hydra_get_service_list_token: "maiquauzuwooQu9ooR7x"
```

2.2.5.1.8 Настройка параметров jwt_key

Настройка параметров приведена в таблице 26.

Таблица 26 – Настройка параметров jwt_key

Параметр	Тип	Описание
jwt_key:		Параметры **jwt_key**
priv:	str	Закрытый ключ
pub:	str	Публичный ключ

Пример корректно настроенного параметра:

```
jwt_key:
  priv: |
    -----BEGIN RSA PRIVATE KEY-----
    MIIEowIBAAKCAQEA063xN82YOtJBq8sfd79bJ+4W9QEdOueQljPziN4JdYntS381
    AqxOA4Ud886S4LdwCN2KSpuh7QSHkzjzH9AlDrHCYShOfPAeHLMCBDszazr2IOc0
    Jaw3bHRfrM9Ilb+X4qdDE88Mfk+B/8Sa/xG2HJVy0Jjb4XoipwzEB900a+6zpnLT
    ..............
    q/kNt7YtrTBmrgpSzFMr0VD+x0Ftb9zhiFVLyVscVKEl67+TNlWahgzh14YF8xP8
    gb89coH114YUNfxN8lKURdY9QFNuZLF+x8xfL4CWwydSbtL7dFFK0HVowMt4tnoJ
    okthJ5JZNw+XZAXHS3NyuvbYmP+iqRuL2YAr2wIDAQABAoIBAGyNHs5HGHHRsOuw
    Uq3/k9aD8NKVjJnJ7/kQEnLlBjcHcpazMHQJnvpfRaqfBre0G1ok9sPH/rvTgK1U
    clKH2eSXgRhKgLf3Dtf6m2bULj0HN0FIydngh0F1EqK10vnvqfkN
    -----END RSA PRIVATE KEY-----
  pub: |
    -----BEGIN PUBLIC KEY-----
    MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA063xN82YOtJBq8sfd79b
    J+4W9QEdOueQljPziN4JdYntS381AqxOA4Ud886S4LdwCN2KSpuh7QSHkzjzH9Al
    DrHCYShOfPAeHLMCBDszazr2IOc0Jaw3bHRfrM9Ilb+X4qdDE88Mfk+B/8Sa/xG2
    HJVy0Jjb4XoipwzEB900a+6zpnLTq/kNt7YtrTBmrgpSzFMr0VD+x0Ftb9zhiFVL
    yVscVKEl67+TNlWahgzh14YF8xP8gb89coH114YUNfxN8lKURdY9QFNuZLF+x8xf
    L4CWwydSbtL7dFFK0HVowMt4tnoJokthJ5JZNw+XZAXHS3NyuvbYmP+iqRuL2YAr
    2wIDAQAB
    -----END PUBLIC KEY-----
```

2.2.5.1.9 Настройка параметров keepalived

Настройка параметров приведена в таблице 27.

Таблица 27 – Настройка параметров keepalived

Параметр	Тип	Описание
keepalived_vrrp_instances		Параметры **keepalived**
ucs_frontend:		Параметры **ucs_frontend**
password:	str	Пароль
virtual_ip	str	Виртуальный IP для группы хостов **ucs_frontend**
ucs_mail:		Параметры **ucs_mail**
password:	str	Пароль
virtual_ip	str	Виртуальный IP для группы хостов **ucs_mail**

Пример корректно настроенного параметра:

```
keepalived_vrrp_instances:
  ucs_frontend:
    password: "UgohSh8i"
    virtual_ip: "192.168.10.10"
    ucs_mail:
      password: "keeB5ooH"
      virtual_ip: "192.168.10.10"
```

2.2.5.1.10 Настройка параметров mailion

Настройка параметров приведена в таблице 28.

Таблица 28 – Настройка параметров mailion

Параметр	Тип	Описание
mailion_cluster:	bool	Флаг кластерной или «Standalone» инсталляции
mailion_domain_module	special	Переменная для генерации эндпоинтов инсталляции (убедиться, что в значении используются разделители «-», а не «.»)
mailion_external_domain:	str	Внешний домен инсталляции
mailion_installation_admin_password	Str	Пароль для администратора всей инсталляции (!)
mailion_integrations	dict	Словарь, содержащий настройки интеграций
mailion_integrations.microsoft	bool	Включение и отключение интеграции с решениями Microsoft
mailion_integrations.freeipa	bool	Включение и отключение интеграции с FreeIpa
mailion_integrations.squadus	bool	Включение и отключение интеграции с ПО Squadus
mailion_integrations.co_auth	bool	Включение и отключение интеграции с ПО «МойОфис «Частное Облако»
mailion_integrations.psn	bool	Включение и отключение интеграции с PSN

Параметр	Тип	Описание
mailion_integrations.google_oauth	bool	Включение и отключение интеграции с Google OAuth
mailion_internal_web_auth	dict	Словарь, содержащий настройки внутренней веб-аутентификации
mailion_internal_web_auth.enabled	bool	Включение и отключение аутентификации для доступа к веб-интерфейсам инфраструктурных сервисов (мониторинг, grafana и т.д.)
mailion_internal_web_auth.password	str	Пароль для аутентификации для доступа к веб-интерфейсам инфраструктурных сервисов
mailion_max_users:	int	Максимальное количество пользователей в инсталляции
mailion_service_accounts	dict	Словарь, содержащий пароли сервисов (values) и имена сервисов (keys)
mailion_supported_domains	list	Список доменов, которые инсталляция будет поддерживать

Пример корректно настроенного параметра:

```
mailion_cluster: true
mailion_domain_module: "{service}.{domain}"
mailion_external_domain: "installation.example.net"
mailion_installation_admin_password: "oor3Iekichocaiphahr5"
mailion_integrations:
  aldpro: false
  co_auth: false
  freeipa: false
  google_oauth: false
  microsoft: false
  psn: false
  samba_dc: false
  squadus: false
mailion_internal_web_auth:
  enabled: true
  password: "rfkg7shtasjfha6vnd"
mailion_max_users: 100
mailion_service_accounts:
  ararat: "Jo8be1pheicahmieV2oa"
  ares: "72gyV456uh9ARiYs8jBx"
  ariadne: "Um6heiNie2doeshee2sa"
  atlas: "Gaezohg1Ad3naf5ahpef"
```

```
  clotho: "Hyrq5iedwemdLNrV47KT"
  cox: "Ii0eeceen5ti1Oe6xaeb"
  dflink_plugin: "Gaezohg1Ad312345hpef"
  elysion: "le0eelePhooghougoopo"
  erakles: "Ui6ohDahLeitozughugh"
  hog: "shee8einoh4AivigePei"
  homeros: "ooph8Efu1eesu2quah1u"
  house: "ahb9Hai3Quaid4aed7an"
  imap: "feo6aita3El6aiMaeboh"
  kongur: "aa6eizooguPhene9uifu"
  kronos: "iphuTh0eiY2ook4aeph5"
  leda: "72YjiCQrnwUwCR32sVrL"
  lmtp: "aicae3yo7Aukaejeel2e"
  marker: "eer1edaeceeJu6naiPom"
  minos: "eshegh3iaR0fie0G"
  othrys: "eeth8Avohv8OpheeHieg"
  paranoid: "Yoa4eNgahm0aeChu8uWe"
  perseus: "Oogh9ahroow2eicaeng7"
  razor: "Ohquietikenu2Aeloh6E"
  theseus: "eileixietai0cahQu3ma"
  viper: "Feir8uewie4Ieshu4thi"
  woof: "at6Ohdapohaitahtho2j"
  zeus: "fa4Ohxaithee0yae1eit"
mailion_supported_domains: []
```

2.2.5.1.11 Настройка параметров mongodb

Настройка параметров приведена в таблице 29.

Таблица 29 – Настройка параметров MongoDB

Параметр	Тип	Описание
mongodb_root_password:	str	Пароль пользователя root для СУБД
mongodb_secured_key:	str	Ключ для доступа к СУБД
mongodb_management_users:		Словарь. Каждый ключ словаря – пользователь
marker:		Ключ, имя пользователя
database:	str	База для аутентификации (опционально)
password:	str	Пароль для аутентификации
roles:		Список ролей (опционально)
- role:	str	Роль пользователя
db	str	Имя базы данных, для которой пользователю присваивается роль

Пример корректно настроенного параметра:

```
## MongoDB secrets
## Generate the password with `pwgen 16 1`
mongodb_root_password: "ohre4Rohngahshah"
## Generate the password with `pwgen 16 1`
mongodb_secured_key: "uGhie5ieweixae9C"
mongodb_management_users:
  achill:
    password: "cohh0Av2mai2aJae"
  beef:
    password: "idohjie2Ikeice0I"
  clotho:
    password: "wahcoovei0bahRu4"
  daidal:
    password: "cheYichoongoh4gi"
  erakles:
    password: "Uxeu4iephluix1ah"
  hog:
    password: "rae0faeng1Seupee"
  homeros:
    password: "xoopunaihuopae4J"
  marker:
    password: "ohvufoosaeTeeCo3"
  mongodb_exporter:
    password: "woo2Yual2saeboh1"
  kongur:
    password: "ahmeayooH1yahlohreem"
  kronos:
    password: "peiNguxud8ooThaiCahL"
  odusseus:
    password: "oY9ja7ietheec6sahthe"
  perseus:
    password: "xuoboop5Geneemei"
  sophokles:
    password: "baexuli5oow8ohTh"
  talaos:
    password: "Ahroozait4pesupohpho"
  themis:
    password: "feef8euch8gaiwieRoig"
  theseus:
    password: "ua8mu0uoj6uvieDu2gei
  "thoth:
    password: "BooRah6oal9Naehai2ph"
```

2.2.5.1.12 Настройка параметров nats

Настройка параметров приведена в таблице 30.

Таблица 30 – Настройка параметров NATS

Параметр	Тип	Описание
nats_authorization_password:	str	Пароль для авторизации в **NATS**
nats_cluster_authorization_password:	str	Пароль для **NATS cluster auth**

Пример корректно настроенных параметров:

```
nats_authorization_password: "Fiohoogh7Raobi4yeiSi"
nats_cluster_authorization_password: "ao1Iey7luRoh1ahf9eVe"
```

2.2.5.1.13 Настройка дополнительных параметров postfix

Настройка дополнительных параметров приведена в таблице 31.

Таблица 31 – Настройка дополнительных параметров postfix

Параметр	Тип	Описание
postfix_additional_mynetworks:	list	Список дополнительных сетей, из которых разрешена отправка через MTA инсталляции

Если используется Exchange, то нужно добавить адреса в исключение для `postfix_additional_mynetworks` адресов Exchange

Пример корректно настроенного параметра:

```
## POSTFIX configuration
### (optional) list of networks allowed to use this SMTP relay
# postfix_additional_mynetworks:
#  - "192.168.113.0/24"
```

2.2.5.1.14 Настройка параметров redis

Настройка параметров приведена в таблице 32.

Таблица 32 – Настройка параметров redis

Параметр	Тип	Описание
redis_cluster_replicas	int	redis_cluster_replicas аналогичен параметру - replicas redis-cli. см. официальную документацию https://redis.io/docs/manual/replication/. Для HA redis с slave, требуется минимум 6 машин с redis_cluster_replicas 1, и 9 машин с redis_cluster_replicas 2.

Параметр	Тип	Описание
redis_dafnis_password	str	Пароль для **redis_dafnis**
redis_dowal_password	str	Пароль для **redis_dowal**
redis_ektor_password	str	Пароль для **redis_ektor**
redis_erakles_password	str	Пароль для **redis_erakles**
redis_euripides_password	str	Пароль для **redis_euripides**
redis_hog_password	str	Пароль для **redis_hog**
redis_homeros_password	str	Пароль для **redis_homeros**
redis_leda_password	str	Пароль для **redis_leda**
redis_minos_password	str	Пароль для **redis_minos**
redis_rspamd_password	str	Пароль для **redis_rspamd**
redis_sdd_password	str	Пароль для **redis_sdd**
redis_viper_password	str	Пароль для **redis_viper**

Пример корректно настроенных параметров:

```
redis_dafnis_password: "eexaiSheQuoivu1oo4ak"
redis_dowal_password: "oasu7nieNg0aeshaiphi"
redis_ektor_password: "eisach9eet8thaug9Ieg"
redis_erakles_password: "zae9iaL3ooth3ahphugh"
redis_euripides_password: "xi6Ohy8io5ku7veQuau7"
redis_hog_password: "dighaeX0hoov6aeJee3u"
redis_homeros_password: "chae7quah7Li2zohbe8o"
redis_leda_password: "Aiy6iiyeiZo2caaleofe"
redis_minos_password: "quie2jiG2CeucooShahG"
redis_rspamd_password: "Iughoo2iuS2Xew1die4p"
redis_sdd_password: "fohphow6eat1aekod5Oh"
redis_viper_password: "Tee9han6ienaYoSievoo"
```

2.2.5.1.15 Настройка параметров resolv

Настройка параметров приведена в таблице 33.

Таблица 33 – Настройка параметров resolv

Параметр	Тип	Описание
resolv_nameservers:	list	Список строк с адресами DNS-серверов для настройки файла **resolv.conf**

Пример корректно настроенного параметра:

```
resolv_nameservers:
  - "192.168.1.1"
  - "192.168.1.2"
  - "192.168.1.3"
```

2.2.5.1.16 Настройка параметров rspamd

Настройка параметров приведена в таблице 34.

Таблица 34 – Настройка параметров rspamd

Параметр	Тип	Описание
rspamd_dkim_hosts:		Параметры антиспама
		Параметры **dkim_hosts**
<your_external_domain>		Имя внешнего домена, который необходимо подписывать DKIM-ключом
dkim_key:	str	DKIM-ключ
rspamd_web_password:	str	Пароль от веб-интерфейса

Пример корректно настроенного параметра:

```
rspamd_dkim_hosts:
    installation.example.net:
      dkim_key: |
        -----BEGIN PRIVATE KEY-----
        MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQC3euVQm/Djy1z1
        JhbTC5Cs99HmrgN6DldM5xivTyhopgkGlHXIoWaKfvt3wKm/Pzah2/BkcTXtDa3w
        E70bmjVXFX2xkXG5DAuY9ChnX6+xWYCeBUeRsMSnWdyoNBwFK9rjE2vZ+u3OzLhz
        wP6PuIyigV7A3D9Mtok0XA3iH/7G+99ARjxhj8hCkYEqEsR688uU1JNeztTfkte+
        mz6n7w8AO2jdpdG8wRqjvj4B4H0MaaP7R4y/UopZ+UP0RAbm7KryOjgC15uLou9Y
        Yg9ym0VkcaIOvc0xQT7Zk13yf8vIuVS/6yhO3FcKYB4mx0Szz1RpU2ueyvD2COSj
        C+2uZsPFAgMBAAECggEBAK6+xEH2kwFRAPKWWSydGigySl4KI1OO7wRWIMNuf4zT
        fUsf/+GaHoAPGk7eVozHlq+nOhdfXz2rRpqdIgF06BJNbI2+ePIFj9IXz5dMoZcm
        KAHYA2a1VUYRpr8oCfu+3dRg/dn4S58miRHtoESfPonS7rx9x2e3fYs5lRtk35EA
        Wp5Vy+2U36cKIJLVtAOvzRbGl9SLjPAvuc/WKGda2lA7HBlhep/Yrm0RUoH//5Px
        fJwLVSy34B31Fx1wZk80aVquXCv644GbR89RIQttziHg9q4g/wyZ5/+ZG/967kim
        tKDS8PWhAK5pjUHS9cED/hjs+ITlNCI4qKf2zj2XSqECgYEA3X9zmzAw9JLnelZN
        3oVM/boqtwfPgnO6Y98inDMsecAICWLCAsEsWYY90IB3FQCXJXVrGTxKnHa3S0fR
        MTX5xx3Rta5SWf88jUQCmZEuxHBeIEN9JKebKC97rKIlIJmYJ8PVZ8c6LAvMgmYc
        sd+GyjJAmV+N7j5Eo8tXmZCCuK0CgYEA1A9t7P6GjXQQFrIk8l+x+0JHmIN+DPKs
        eyR6avDfd32HIq2dPCMmjCAl7EFbfOPVNx9rZvLrEWtTkgU8DYBPlZ955EJORi9l
        eqYeOKwhWLUMgwHyWlEJZPeY3o31TF1NwNG16Qy98h4zr2SUaTuCdccoNWAc0GuI
        rA1Gjn7AonkCgYEAuMpgFJS8Aw+cdwARrxffB7+Na23kvZz3X+ME6PP4owqGqe3u
        loW7DmVkpNLIhokbkHDJjSAzzx1sBi5AZKH3ZRuHnd91bQf36JNY5+2r6s8keB5W
        BYKfe4NB1uDfwLbjrik/nXklGyIs2I2AWxV1SrNqGYsSyjTA5zX6O2/I33ECgYBS
        eO23jgWmXc0kBoR4Ym9F2LEfj4QmZPrPqZAypxtBzYAQ7JSKHuGO/bHCAGkkWTdD
        COUsVK03SRZnY8HHPm+1MSCmtWLbyPMekByQzeDqLv9+s/MdTQbqTaEWbP9Jg8AJ
        jYXB7UKyNyzCucs+YfaK97mbiJWsOSYeQ8t8/67LgQKBgQCk4q/D5Cq5Fqalbk/0
        jyEQAmHgrhWEJO2bECGjGIJ13/Hj3bbQ3znfPUDf9MLDtrveGu4YdspL3S4yahLO
        EXxXPgwHCDLqamx5vj4QKFPFQEHXv68RK6RKhw7m2IeyI/7nsHPvjZhNZI4ulSTN
        CLCjuiw8tvIafY26wKDylpvnRQ==
        -----END PRIVATE KEY-----

  rspamd_web_password: "iePixieTaf4IriequieX"
```

2.2.5.1.17 Настройка параметров servus

Настройка параметров приведена в таблице 35.

Таблица 35 – Настройка параметров servus

Параметр	Тип	Описание
servus:	str	Параметры **servus**

Пример корректно настроенного параметра:

```
servus: "Iefae4yoh4rohceepoli"
```

2.2.5.1.18 Настройка параметров sophokles

Настройка параметров приведена в таблице 36.

Таблица 36 – Настройка параметров sophokles

Параметр	Тип	Описание
sophokles_access_token:	str	Токен для сервиса авторизации **minos** и **sophokles**

Пример корректно настроенного параметра:

```
sophokles_access_token: "IeWoh9eateihuvoxekah":
```

2.2.5.1.19 Настройка параметров theseus

Настройка параметров приведена в таблице 37.

Таблица 37 – Настройка параметров theseus

Параметр	Тип	Описание
theseus_cipher_key:	str	Ключ шифрования **theseus**

Пример корректно настроенного параметра:

```
theseus_cipher_key: "RWVmb21pZXhvbmFpYzZvaHlhaTR6aURhd2VpZzhlZW4="
```

2.2.5.1.20 Настройка параметров unbound

Настройка параметров приведена в таблице 38.

Таблица 38 – Настройка параметров unbound

Параметр	Тип	Описание
unbound_access_control:	dict	Параметры доступа к управлению **unbound**
network1	str	Подсеть, из которой разрешен доступ к кэширующему **DNS**
unbound_enable_autowildcard:	bool	Флаг использования автоматического формирования DNS-записей внутренних адресов на базе серверов в файле **inventory** и их значений переменной **ansible_default_ipv4**

Параметр	Тип	Описание
unbound_forward_addresses:	list	Список строк внешних DNS-сервисов, на которые будут перенаправляться запросы **unbound** серверов

Пример корректно настроенного параметра:

```
unbound_enable_autowildcard: false
unbound_access_control:
  network1: "192.168.1.0/24"
unbound_forward_addresses:
  - "8.8.8.8"
  - "1.1.1.1"
```

2.2.5.1.21 Настройка параметров CA

Настройка параметров приведена в таблице 39.

Таблица 39 – Настройка параметров CA

Параметр	Тип	Описание
tls_certs_remote_token_key:	str	Ключ для доступа к API внутреннего **Certificate Authority**

Пример корректно настроенного параметра:

```
tls_certs_remote_token_key: "afba15d0def55ca6e57efb481f8232a5"
```

2.2.5.1.22 Настройка параметров viper

Настройка параметров приведена в таблице 40.

Таблица 40 – Настройка параметров viper

Параметр	Тип	Описание
viper_calendar_settings_sender_white_list:	list	Список отправителей, которые могут присылать календарные письма за участника события (например, сервисные ящики noreply).
regexp:	str	Выбор отправителей по регулярному выражению
sender_in_reply_to	bool	Добавить отправителя в **reply_to**
Переменные для ограничения индексации писем		
viper_rate_limit_mail_indexer_enable	bool	Если **true**, то механизм ограничения индексации включен
viper_rate_limit_mail_indexer_events_per_sec	int	Ограничение на количество запросов в секунду (RPS)

Параметр	Тип	Описание
viper_rate_limit_mail_indexer_burst	int	Размер разрешенного единовременного всплеска событий. Данная переменная нужна для обработки пиковой нагрузки. Значение может быть больше ограничения на количество запросов в секунду (RPS). Практическое применение этого параметра заключается в ограничении количества одновременно обрабатываемых событий
viper_rate_limit_mail_indexer_max_delay_sec	int	Максимальное время, на которое может блокироваться обработка события при реиндексации. Если это время превышено, попытка индексации будет происходить позже, когда нагрузка станет меньше. Для отключения проверки максимальной задержки следует установить значение 0
Переменные для ограничения индексации вложений писем		
viper_rate_limit_attachment_indexer_enable	bool	Если **true**, то механизм ограничения индексации включен
viper_rate_limit_attachment_indexer_events_per_sec	int	Ограничение на количество запросов в секунду (RPS)
viper_rate_limit_attachment_indexer_burst	int	Размер разрешенного единовременного всплеска событий. Данная переменная нужна для обработки пиковой нагрузки. Значение может быть больше ограничения на количество запросов в секунду (RPS). Практическое применение этого параметра заключается в ограничении количества одновременно обрабатываемых событий
viper_rate_limit_attachment_indexer_max_delay_sec	int	Максимальное время, на которое может блокироваться обработка события при реиндексации. Если это время превышено, попытка индексации будет происходить позже, когда нагрузка станет меньше. Для отключения проверки максимальной задержки следует установить значение 0

Пример корректно настроенного параметра:

```
viper_calendar_settings_sender_white_list:
- regexp: "[^@]+@calendar.example.ru$"
  sender_in_reply_to: true
- regexp: "^calendar@calendar.example.ru$"
  sender_in_reply_to: false

viper_rate_limit_mail_indexer_enable: false
viper_rate_limit_mail_indexer_events_per_sec: 100
viper_rate_limit_mail_indexer_burst: 50
viper_rate_limit_mail_indexer_max_delay_sec: 300
```

```
viper_rate_limit_attachment_indexer_enable: false
viper_rate_limit_attachment_indexer_events_per_sec: 100
viper_rate_limit_attachment_indexer_burst: 50
viper_rate_limit_attachment_indexer_max_delay_sec: 300
```

2.2.5.1.23 Настройка параметров ntp

Настройка параметров приведена в таблице 41.

Таблица 41 – Настройка параметров ntp

Параметр	Тип	Описание
ntp_servers:	list	Список **ntp** серверов
ntp_listen_on_default_v4	bool	Определяет какие сетевые адреса открывает **ntpd**
ntp_listen_on_default_v6	bool	Определяет какие сетевые адреса открывает **ntpd**
ntp_clients_inventory_access	bool	Ограничивает все хосты из inventory по флагу **nomodify notrap**
ntp_clients:	list	Список хостов/адресов для ограничения
name:	str	Имя хоста или адреса
access:	str	Флаг доступа
ntp_driftfile_directory	str	Путь к файлу данных **ntp**
ntp_custom_config	dict	Выборочная конфигурация **ntp**

2.2.5.1.24 Настройка параметров chrony

Настройка параметров приведена в таблице 42.

Таблица 42 – Настройка параметров chrony

Параметр	Тип	Описание
ntp_servers:	list	Список **ntp** серверов

2.2.5.1.25 Настройка учетных записей администраторов

Необходимо настроить учетные данные дополнительных администраторов (помимо администратора инсталляции (AdminInstallation) и администратора тенанта (AdminTenant)):

- администратора аудита (AdminAudit);
- администратора информационной безопасности (AdminInformationSecurity);
- суперадминистратора (AdminSuper).

Рекомендация от ИБ

Для суперадминистратора рекомендуется установить пароль, состоящий из двух частей, известных двум другим администраторам, чтобы действия этого администратора требовали подтверждения двух других.

При переходе на версию Mailion 2.0 выполняется миграция прав администраторов с ролями AdminInstallation и AdminTenant. В будущих релизах возможны изменения в наборах прав для администраторов со стандартными ролями AdminInstallation, AdminTenant, AdminSuper, AdminAudit, AdminAuditTenant, AdminInformationSecurity. Поэтому для расширения возможностей администраторов можно создавать отдельные роли с настраиваемыми наборами прав (т. е. настраиваемых администраторов).

Описание параметров приведено в таблице 43.

Таблица 43 — Учетные данные администраторов

Параметр	Тип	Описание
mailion_audit_admin_login	str	Логин администратора аудита (по умолчанию "audit_admin")
mailion_audit_admin_password	str	Пароль администратора аудита
mailion_information_security_admin_login	str	Логин администратора информационной безопасности (по умолчанию "is_admin")
mailion_information_security_admin_password	str	Пароль администратора информационной безопасности
mailion_super_admin_login	str	Логин суперадминистратора (по умолчанию "super_admin")
mailion_super_admin_password	str	Пароль суперадминистратора

2.2.6 Настройка межсетевого экранирования

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

Установленные правила межсетевого экрана приведены в таблице 44.

Таблица 44 – Установленные правила межсетевого экрана

Серверы	Таблица	Состояние	Порт	Протокол	Интерфейс	Разрешение
Серверы группы ucs	INPUT					DROP
	FORWARD					DROP
	OUTPUT					ACCEPT
	INPUT	RELATED, ESTABLISHED				ACCEPT
	INPUT			ICMP		ACCEPT

Серверы	Таблица	Состояние	Порт	Протокол	Интерфейс	Разрешение
	INPUT				lo	ACCEPT
	INPUT	NEW	SSH	TCP		ACCEPT
	INPUT				docker0 или cni-podman0*	ACCEPT
	INPUT		2376	TCP		ACCEPT
	INPUT		9100	TCP		ACCEPT
Серверы группы ucs_etcd	INPUT					DROP
	FORWARD					DROP
	OUTPUT					ACCEPT
	INPUT	RELATED, ESTABLISHED				ACCEPT
	INPUT			ICMP		ACCEPT
	INPUT				lo	ACCEPT
	INPUT	NEW	SSH	TCP		ACCEPT
	INPUT				docker0 или cni-podman0*	ACCEPT
	INPUT		2376	TCP		ACCEPT
	INPUT		9100	TCP		ACCEPT
	INPUT	NEW	53	UDP		ACCEPT
Серверы группы ucs_infrastructure	INPUT					DROP
	FORWARD					DROP
	OUTPUT					ACCEPT
	INPUT	RELATED, ESTABLISHED				ACCEPT
	INPUT			ICMP		ACCEPT
	INPUT				lo	ACCEPT
	INPUT	NEW	SSH	TCP		ACCEPT
					docker0 или cni-podman0*	ACCEPT
	INPUT		2376	TCP		ACCEPT
	INPUT		9100	TCP		ACCEPT
	INPUT		53	TCP		ACCEPT
	INPUT		53	UDP		ACCEPT
Серверы группы ucs_frontend	INPUT					DROP
	FORWARD					DROP
	OUTPUT					ACCEPT
	INPUT	RELATED, ESTABLISHED				ACCEPT
	INPUT			ICMP		ACCEPT
	INPUT				lo	ACCEPT

Серверы	Таблица	Состояние	Порт	Протокол	Интерфейс	Разрешение
	INPUT	NEW	SSH	TCP		ACCEPT
					docker0 или cni-podman0*	ACCEPT
	INPUT		2376	TCP		ACCEPT
	INPUT		9100	TCP		ACCEPT
	INPUT			VRRP		ACCEPT
Серверы группы ucs_mail	INPUT					DROP
	FORWARD					DROP
	OUTPUT					ACCEPT
	INPUT	RELATED, ESTABLISHED				ACCEPT
	INPUT			ICMP		ACCEPT
	INPUT				lo	ACCEPT
	INPUT	NEW	SSH	TCP		ACCEPT
					docker0 или cni-podman0*	ACCEPT
	INPUT		2376	TCP		ACCEPT
	INPUT		9100	TCP		ACCEPT
	INPUT			VRRP		ACCEPT

***** если используется контейнерная виртуализация **podman**.

2.2.6.1 Настройки правил внешнего межсетевого экрана

Во время установки происходит настройка межсетевого экрана внутри контура инсталляции. Тем не менее, очень важно обеспечить дополнительную защиту системы с внешней стороны по отношению к контуру инсталляции.

Во внешний контур должны быть доступны только следующие порты:

– порты на виртуальные IP серверов с ролью **ucs_frontend**:

- 80/tcp;
- 143/tcp;
- 443/tcp;
- 993/tcp;
- 3142/tcp;
- 6787/tcp;

- 389/tcp;
- 389/udp;
- 636/tcp;
- 636/udp;

– порты на виртуальные IP серверов с ролью **ucs_mail**:

- 465/tcp;
- 587/tcp;

– порты на реальные IP серверов **ucs_mail**:

- 25/tcp.

2.3 Запуск установки

Для установки на контейнеры **docker** необходимо запустить следующую команду:

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

Для установки на контейнеры **podman** необходимо запустить следующую команду:

```
ansible-playbook playbooks/main.yml --skip-tags=cadvisor --extra-vars
'{"container_management_tool": "podman"}'  --extra-vars
'{"podman_container_no_hosts": "true"}' -e '{"confd_max_memory": "100M"}' -e
'{"pregen_max_memory": "100M"}' -e '{"cvm_max_memory": "2000M"}' --diff
```

Если использовалось шифрование паролей, описанное в разделе Установка Mailion с машины оператора, то к команде установки необходимо добавить ключи `--vault-password-file=.filesecret` или `--ask-vault-pass`.

После этого запускаются роли, описанные в разделе Описание ролей Ansible.

После установки в Mailion по умолчанию включен повышенный уровень журналирования. Из-за этого дисковое пространство будет быстро заполняться данными журналов. Чтобы отключить эту функцию, необходимо на машине оператора в файле `~/install_mailion/group_vars/ucs/main.yml` в строке 1888 удалить параметр `syslog_ng_external_all_services: true`. Сделать это можно с помощью следующей команды:

```
sed 's|syslog_ng_external_all_services: true|
#syslog_ng_external_all_services: true|g' -i
~/install_mailion/group_vars/ucs/main.yml
```

Затем необходимо перезапустить сервис **syslog_ng**:

```
~/install_mailion# ansible-playbook playbooks/mailion/logging.yml -v
```

2.4 Проверка корректности установки

Для проверки корректности установки необходимо запустить установленный ПО «Mailion»:

1. Открыть в поддерживаемом веб-браузере страницу по адресу, который указывался в **mailion_external_domain**.
2. Использовать для входа учетные данные созданных пользователей.
3. Если вход был выполнен под пользователем, то необходимо отправить письмо самому себе внутри ПО «Mailion». Если вход выполнен под администратором, то сначала нужно создать пользователя (при условии, что он не был создан плейбуком **ministerium**).
4. Если письмо успешно отправилось и пришло — установка настроена корректно.

2.4.1 Добавление дополнительных доменов для обслуживания инсталляцией

В ПО «Mailion» реализована поддержка дополнительных доменов. Чтобы добавить дополнительный домен необходимо включить его в список **mailion_supported_domain**:

```
mailion_supported_domains:
  - "example.com"
```

Затем необходимо добавить dkim-ключ к домену в словарь **rspamd_dkim_hosts**:

```
rspamd_dkim_hosts:
  domain2.example.net:
    dkim_key: |
......
```

После этого с машины оператора из папки с инсталлятором необходимо выполнить команду:

```
ansible-playbook playbooks/ucs/main.yml --tags postfix,rspamd --limit ucs_mail --
diff
```

Эта команда запустит роль **postfix** с функцией **mx** и добавит указанные домены для **MTA**, а также добавит dkim-ключи для доменов в **rspamd**.

2.5 Установка в составе других продуктов ПО «МойОфис»

Установка в составе других продуктов ПО «МойОфис» не выполняется.

3 ОБНОВЛЕНИЕ С ПРЕДЫДУЩИХ ВЕРСИЙ

3.1 Процедура обновления Mongodb с версии 4.4.10-17 до 6.0.14-32

Перед обновлением Mailion до версии 2.0 необходимо обновить Mongodb до версии 6.0.14-32. Перед обновлением необходимо проверить наличие достаточного свободного места на дисках серверов группы **db**. Процедура может занимать достаточно долгое время, которое зависит от объема конкретной базы и скорости дисков. Например для базы объемом 130 Гбайт на SSD-дисках этот процесс занял около 14 часов. Команды ansible и плейбуки выполняются на машине оператора.

1. На время обновления Mongodb необходимо остановить все сервисы Mailion:
   - Выполнить команду:

```
ansible-playbook -i hosts.yml
playbooks/mailion/disaster/disaster_stop_services.yml
```

   - Если при выполнении плейбука возникает ошибка:

```
failed: [ucs-search-2.example.com] (item=helpbek) => {"ansible_loop_var": "item",
"changed": false, "item": "helpbek", "msg": "Cannot create container when image is
not specified!"}
```

     необходимо удалить из плейбука указанный сервис (в примере это **helpbek**):

```
playbooks/mailion/disaster/disaster_stop_services.yml
```

     и заново запустить плейбук.
   - Проверить через Kunkka (http://kunkka.ucs-infra-1.example.com/), что сервисы остановлены (остаются запущенными только **dispersed-pbject-store, dirbek, mailbek-search**).
2. Создать резервную копию Mongodb, используя **mongodump** версии **4.4.10-17** с помощью скрипта, включенного в базовую поставку. Скрипт находится на машине группы **infra** по пути `/srv/docker/mongodb/backup_scripts/mongodb_backup_generic.sh`. Перед запуском необходимо проверить наличие достаточного свободного места под резервную копию в директории `/srv/backups`.
3. После успешного создания резервной копии остановить все контейнеры MongoDB с помощью команды:

```
ansible -i hosts.yml ucs_mongodb -m ansible.builtin.shell -a "sudo docker stop
mongodb_generic"
```

4. Так как была создана резервная копия базы, то для экономии дискового пространства следует почистить данные базы, выполнив следующую команду:

```
ansible -i hosts.yml ucs_mongodb -m ansible.builtin.shell -a "sudo rm -
r /srv/docker/mongodb_generic/ || echo 'Already deleted'"
```

Если на серверах группы **db** достаточно дискового пространства, то можно вместо команды `rm` выполнить команду `mv`, что позволит в случае каких либо проблем с обновлением быстро выполнить откат к прежней версии. В этом случае следует выполнить команду:

```
ansible -i hosts.yml ucs_mongodb -m ansible.builtin.shell -a "sudo
mv /srv/docker/mongodb_generic/ /srv/docker/mongodb_generic.bak || echo 'Already
moved'"
```

5. Установить Mongodb версии 6.0.14-32 с помощью плейбука Ansible.
   - Для этого необходимо убедится что на машине оператора установлены плейбуки из релиза Mailion 2.0. В файле `group_vars/ucs/versions.yml` переменная **mongodb_image_tag** должна иметь следующее значение:

```
mongodb_image_tag: "6.0.14-32"
```

   - Выполнить команду

```
ansible-playbook playbooks/mailion/infra.yml -i hosts.yml \
        --diff \
        --tags mongodb
```

   - Проверить, что образ контейнера Mongodb действительно имеет версию **6.0.14-32**.
     – Зайти на любую машину группы **db**.
     – Выполнить следующую команду и проверить в выводе версию образа:

```
sudo docker ps -a | grep mongodb
```

6. Выполнить восстановление данных в кластер **Mongodb** новой версии 6.0.14-32. Пример скрипта для запуска восстановления данных:

```
#!/bin/bash -xe
MONGO_VERSION="6.0.14"
BACKUP_FILE="mongodb_dump_2024_08_27_1525.gz"
BACKUP_DIR="/srv/backups/mongodb/"
MONGODB_ROOT_PASSWORD="<MONGODB_ROOT_PASSWORD>"
DB_LIST=(
    "achill" \
    "beef" \
    "clotho" \
    "dafnis" \
    "daidal" \
    "dorofej" \
    "ektor" \
    "erakles" \
    "eratosthenis" \
    "euripides" \
    "hog" \
    "homeros" \
    "kongur" \
    "kronos" \
    "marker" \
    "odusseus" \
    "perseus" \
    "sophokles" \
    "talaos" \
    "themis" \
    "theseus" \
    "thoth"
)for db in ${DB_LIST[@]}; do
        echo $db;
        docker run -it --rm -v ${BACKUP_DIR}:/backups -
v /srv/tls/certs:/etc/pki/tls/certs \
        -e "MONGO_CONN=mongodb://root:${MONGODB_ROOT_PASSWORD}@@mongodb.ucs-db-
1.example.com:27017,mongodb.ucs-db-2.example.com:27017,mongodb.ucs-db-
3.example.com:2701/?
authSource=admin&replicaSet=ucs&tls=true&tlsCAFile=/etc/pki/tls/certs/ucs-db-
1.example.com-main-ca.pem&tlsCertificateKeyFile=/etc/pki/tls/certs/merged_ucs-db-
1.example.com-main-peer.pem" \
        --name mongorestore hub.example.com/mongo:6.0.14-32 \
        sh -c "mongorestore --drop  --gzip \$MONGO_CONN --
authenticationDatabase=admin --nsInclude=\"$db.*\" --numParallelCollections=1 --
maintainInsertionOrder --numInsertionWorkersPerCollection=1 --objcheck --
convertLegacyIndexes --stopOnError  --archive=/backups/${BACKUP_FILE}"
done
```

В этом скрипте при необходимости поменять хосты, сертификаты, путь до файла резервной копии и пароли от базы **Mongodb**. Файл сертификата `merged_mongodb.ucs-db-1.example.com-main-peer.pem`, находящийся по пути `/srv/tls/certs/`, необходимо скопировать с любой ноды группы **db** (в нашем примере с первой) на машину группы **infra**.

7. Проверить, что данные восстановлены. Для этого:
   - Подключиться к **Mongodb** любым удобным способом.
   - Выполнить команды:

```
use admin;
show dbs;
```

8. Запустить сервисы Mailion:
   - Выполнить команду:

```
ansible-playbook \
    -i hosts.yml playbooks/mailion/disaster/disaster_start_services.yml
```

- Если при выполнении плейбука возникает ошибка

```
failed: [ucs-search-2.example.com] (item=helpbek) => {"ansible_loop_var": "item",
"changed": false, "item": "helpbek", "msg": "Cannot create container when image is
not specified!"}
```

– Закомментировать в плейбуке указанные сервисы (в примере это **helpbek**)

```
playbooks/mailion/disaster/disaster_start_services.yml
```

– Заново запустить плейбук

- Проверить через Kunkka (http://kunkka.ucs-infra-1.example.com/), что сервисы запустились.

3.1.1 Решение проблемы с авторизацией некоторых пользователей

В Mailion 2.0 после обновления **mongodb** до версии 6 некоторые делегированные (внешние) пользователи могут столкнуться с невозможностью авторизации в Mailion. Это может быть связано с тем, что при восстановлении дампа базы данных версии 4 в новой версии изменился алгоритм сортировки данных **И** в базе данных присутствовали записи **entity** со статусом 7 (CREATED) с одинаковым идентификатором **external_id**.

Чтобы избежать такой ситуации, необходимо для каждой из реплик базы данных выполнить следующий запрос:

```
use erakles;
db.entities.aggregate(
[
  { $match: { status: { $ne: 5 } } },
  {
    $group: {
      _id: "$external_catalog_data.external_id",
      count: { $sum: 1 },
      entity: {
        $first: {
          entity_id: "$_id",
          status: "$status",
          external_id:
            "$external_catalog_data.external_id"
        }
      }
    }
  },
  {
    $match: {
      count: { $gte: 2 },
      "entity.status": 7
    }
  }
]
)
```

Этот код позволит найти «проблемные» записи **entity**.

Например:

```
{ "_id" : "95f059b2-d4d0-11ee-b9ff-0242ac110002", "count" : 3, "entity" :
{ "entity_id" : UUID("3b69d923-ae84-4785-b6bc-1f0eb2dd5301"), "status" : 7,
"external_id" : "95f059b2-d4d0-11ee-b9ff-0242ac110002" } }
{ "_id" : "f04a7a28-d4d0-11ee-831e-0242ac110002", "count" : 2, "entity" :
{ "entity_id" : UUID("158c24b1-48bb-44e9-8804-083d665648c1"), "status" : 7,
"external_id" : "f04a7a28-d4d0-11ee-831e-0242ac110002" } }
{ "_id" : "64a415bc-d4ce-11ee-ada8-0242ac110002", "count" : 3, "entity" :
{ "entity_id" : UUID("d1fd4633-acba-47c8-bef4-66aec5c6aef5"), "status" : 7,
"external_id" : "64a415bc-d4ce-11ee-ada8-0242ac110002" } }
{ "_id" : "9f7a287a-d532-11ee-8431-0242ac110002", "count" : 2, "entity" :
{ "entity_id" : UUID("31d94611-8d87-47df-bc36-b55bed204951"), "status" : 7,
"external_id" : "9f7a287a-d532-11ee-8431-0242ac110002" } }
```

Затем все найденные записи необходимо удалить по их **entity_id** с помощью **ministerium** (команда `change_status`) или через панель администрирования в веб-интерфейсе Mailion (**entity_id** можно вставить в ссылку в адресной строке браузера, затем удалить найденную запись).

Повторять запрос необходимо на каждой реплике до тех пор, пока не будет возвращен пустой результат.

3.2 Общая процедура обновления

Обновления возможны с версий Mailion 1.8, 1.8.1, 1.8.2 и 1.9. Список компонентов приведен в разделе Состав дистрибутива.

Перед запуском обновления необходимо обновить версию **mongodb**, согласно описанию в разделе Обновление Mongodb до версии 6.0. Перед обновлением необходимо в файлах `group_vars/ucs_setup/*` проверить наличие новых переменных (где `ucs_setup` — название инсталяции). Новые переменные находятся в файлах `contrib/mailion/cluster/group_vars/ucs_setup/*` (для «Standalone» инсталляции — в `contrib/mailion/standalone/group_vars/ucs_setup/*`). Изменения удобнее всего отследить с помощью команды `vimdiff`.

Если обновление осуществляется с версии Mailion 1.9, то необходимо добавить следующие новые переменные:

```
mailion_audit_admin_password: "e1ONTyE9DvbdYaxLzDTD"
mailion_information_security_admin_password: "Zg8TaINjDBjJQGQiDHA6"
mailion_super_admin_password: "Tgm3DoFmeCcTl4iMdcfy"
```

Переменную `redis_sdd_password` можно удалить.

Обновление ПО «Mailion» осуществляется аналогично установке новой версии (см. раздел Первичная установка) за исключением того, что обновление с версии 1.7 до 1.8 должно производиться с параметром `permission_migration=true`, который используется для смены пользователей и прав доступа к файлам сервисов при переходе на режим `rootful`. При

обновлении Mailion этот параметр нужно применить только один раз. При последующих установках отдельных сервисов его использовать не нужно.

Запуск обновления с версии 1.7 с миграцией данных:

```
ansible-playbook playbooks/main.yml -e "permission_migration=true" --diff
```

Обновление с других версий без миграции данных (см. раздел Запуск установки):

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

> ⚠ Перед обновлением необходимо в файле `contrib/mailion/cluster/group_vars/ucs_<имя_инсталяции>/version.yml` присвоить переменной **mailion_release_name** значение, соответствующее номеру актуального релиза.

3.3 Возможные проблемы после обновления версии docker

На некоторых версиях Astra Linux после обновления пакета docker.io возможно появление проблемы неработоспособности автозагрузки docker-контейнеров некоторых сервисов после перезагрузки сервера или docker-демона. Для исправления этой проблемы рекомендуем выполнить откат версии docker до более стабильной следующей командой:

```
ansible -i hosts.yml ucs -m ansible.builtin.shell -a "apt install -y
docker.io=20.10.2+dfsg1-2astra.se10+ci22 --allow-downgrades" -b
```

4 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ И РЕКОМЕНДАЦИИ ПО УСТАНОВКЕ

4.1 Настройка Redis и Sentinel для работы по TLS

По умолчанию поддержка `redis` и `redis-sentinel` для `tls` отключена. Для включения поддержки `redis` и `redis-sentinel` с `tls` необходимо в файле `./папка_инсталяции/group_vars/ucs_Имя_Стенда/main.yml` задать значение `true` для следующих параметров:

```
//  Включает настройку TLS для подключения к Redis в конфигурационных файлах сервисов
mailion_global_redis_tls: true
//  Включает настройки TLS для Redis
redis_tls_enabled: true
//  Включает настройки TLS для Sentinel
redis_sentinel_tls_enabled: true
//  Включает настройки TLS для подключения к Redis
redis_exporter_tls_enabled: true
```

4.2 Доступ к веб-интерфейсам вспомогательных систем для управления ПО «Mailion»

4.2.1 Rspamd

Rspamd – система управления антиспамом (конфигурация правил рейтингов, история обработки). Веб-интерфейс Rspamd доступен по адресу `http://rspamd.<mail_inventory_hostname>:11334/`.

Где `<mail_inventory_hostname>` - FQDN хоста из группы **ucs_mail** (см. Рисунок 4).

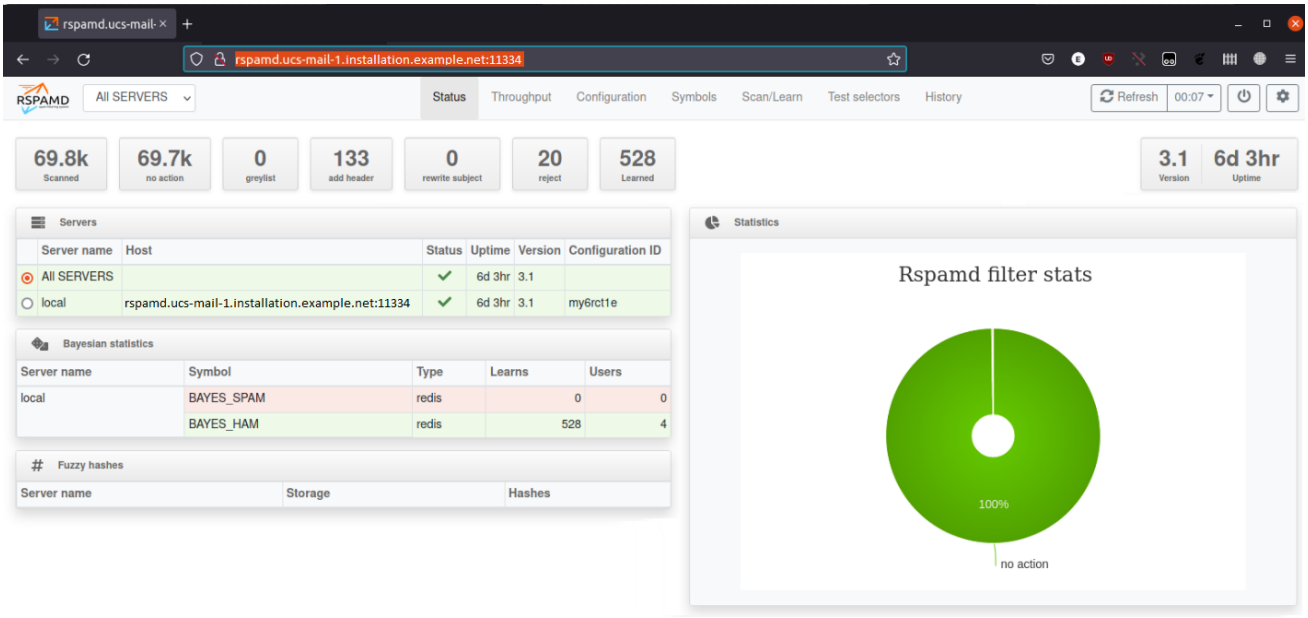


Рисунок 4 – Веб-интерфейс Rspamd

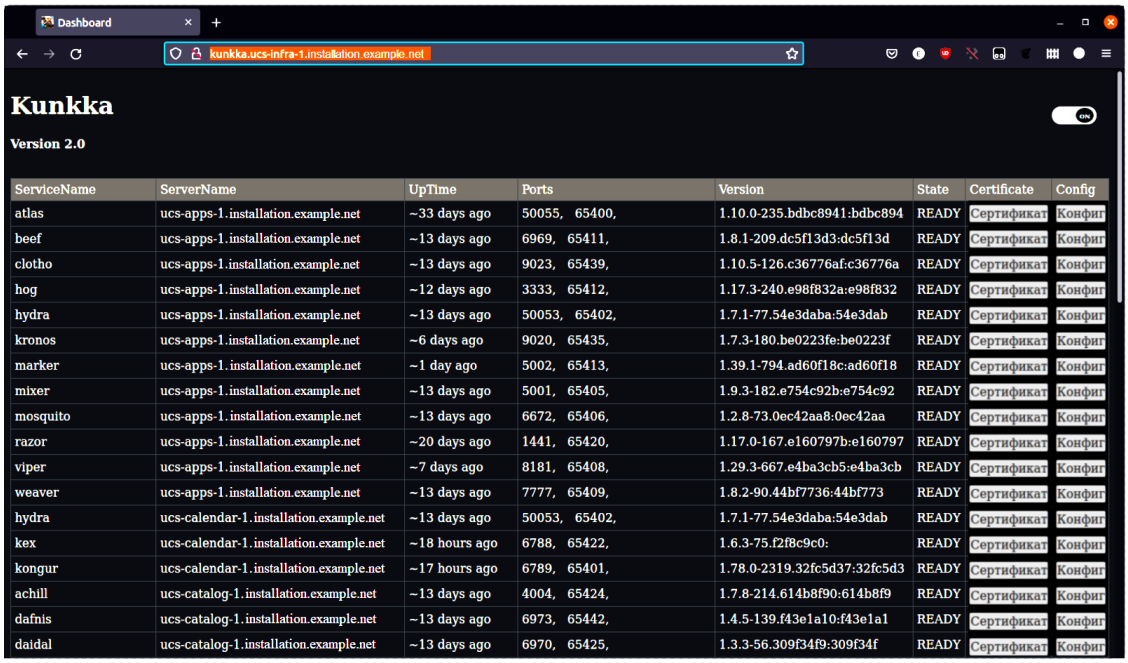
Доступ в Rspamd необходимо осуществлять по протоколу HTTP из внутренней сети инсталляции.

Для доступа к веб-интерфейсу потребуется пароль, который указан в переменной **rspamd_web_password**.

4.2.2 Kunkka

Kunkka – веб-страница с отображением подсистем на серверах. Веб-интерфейс Kunkka доступен по адресу `http://kunkka.<infrastructure_inventory_hostname>/`.

Где `<infrastructure_inventory_hostname>` - FQDN хоста группы **ucs_infrastructure** (см. Рисунок 5).



ServiceName	ServerName	UpTime	Ports	Version	State	Certificate	Config
atlas	ucs-apps-1.installation.example.net	~33 days ago	50055, 65400,	1.10.0-235.bdbc8941:bdbc894	READY	Сертификат	Конфиг
beef	ucs-apps-1.installation.example.net	~13 days ago	6969, 65411,	1.8.1-209.dc5f13d3:dc5f13d	READY	Сертификат	Конфиг
clotho	ucs-apps-1.installation.example.net	~13 days ago	9023, 65439,	1.10.5-126.c36776af:c36776a	READY	Сертификат	Конфиг
hog	ucs-apps-1.installation.example.net	~12 days ago	3333, 65412,	1.17.3-240.e98f832a:e98f832	READY	Сертификат	Конфиг
hydra	ucs-apps-1.installation.example.net	~13 days ago	50053, 65402,	1.7.1-77.54e3daba:54e3dab	READY	Сертификат	Конфиг
kronos	ucs-apps-1.installation.example.net	~6 days ago	9020, 65435,	1.7.3-180.be0223fe:be0223f	READY	Сертификат	Конфиг
marker	ucs-apps-1.installation.example.net	~1 day ago	5002, 65413,	1.39.1-794.ad60f18c:ad60f18	READY	Сертификат	Конфиг
mixer	ucs-apps-1.installation.example.net	~13 days ago	5001, 65405,	1.9.3-182.e754c92b:e754c92	READY	Сертификат	Конфиг
mosquito	ucs-apps-1.installation.example.net	~13 days ago	6672, 65406,	1.2.8-73.0ec42aa8:0ec42aa	READY	Сертификат	Конфиг
razor	ucs-apps-1.installation.example.net	~20 days ago	1441, 65420,	1.17.0-167.e160797b:e160797	READY	Сертификат	Конфиг
viper	ucs-apps-1.installation.example.net	~7 days ago	8181, 65408,	1.29.3-667.e4ba3cb5:e4ba3cb	READY	Сертификат	Конфиг
weaver	ucs-apps-1.installation.example.net	~13 days ago	7777, 65409,	1.8.2-90.44bf7736:44bf773	READY	Сертификат	Конфиг
hydra	ucs-calendar-1.installation.example.net	~13 days ago	50053, 65402,	1.7.1-77.54e3daba:54e3dab	READY	Сертификат	Конфиг
kex	ucs-calendar-1.installation.example.net	~18 hours ago	6788, 65422,	1.6.3-75.f2f8c9c0:	READY	Сертификат	Конфиг
kongur	ucs-calendar-1.installation.example.net	~17 hours ago	6789, 65401,	1.78.0-2319.32fc5d37:32fc5d3	READY	Сертификат	Конфиг
achill	ucs-catalog-1.installation.example.net	~13 days ago	4004, 65424,	1.7.8-214.614b8f90:614b8f9	READY	Сертификат	Конфиг
dafnis	ucs-catalog-1.installation.example.net	~13 days ago	6973, 65442,	1.4.5-139.f43e1a10:f43e1a1	READY	Сертификат	Конфиг
daidal	ucs-catalog-1.installation.example.net	~13 days ago	6970, 65425,	1.3.3-56.309f34f9:309f34f	READY	Сертификат	Конфиг

Рисунок 5 – Веб-интерфейс Kunkka

Для доступа может потребоваться логин «admin» и пароль, хранящийся в переменной **mailion_internal_web_auth.password.**

4.2.3 Prometheus

Prometheus – система мониторинга. Веб-интерфейс Prometheus доступен по адресу `http://prometheus.<infrastructure_inventory_hostname>/`.

Где `<infrastructure_inventory_hostname>` - FQDN хоста группы **ucs_infrastructure** (см. Рисунок 6).

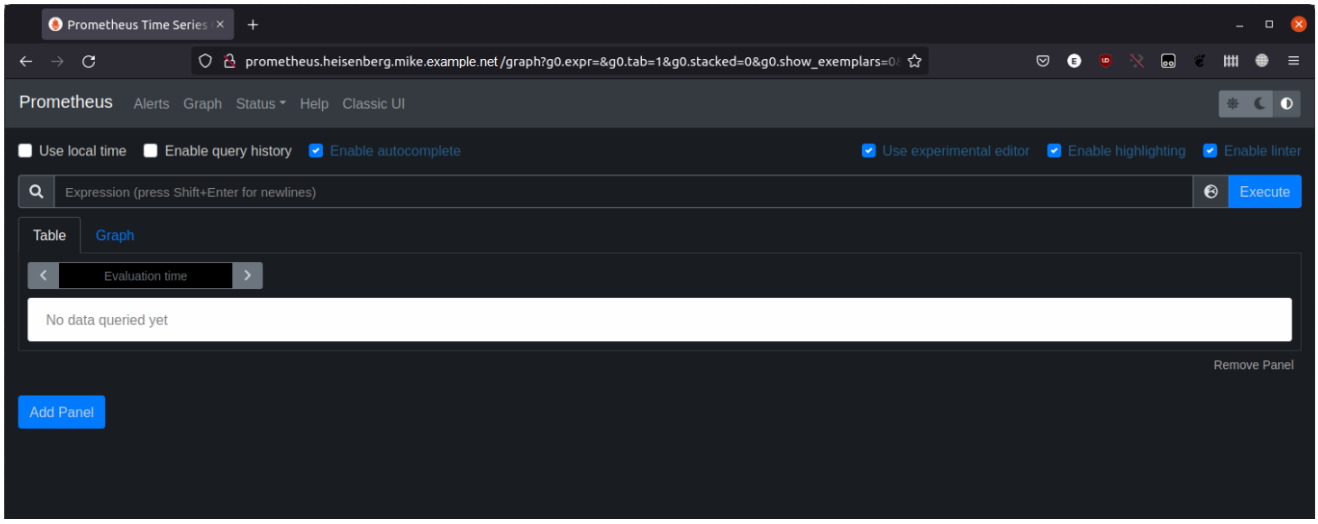


Рисунок 6 – Веб-интерфейс Prometheus

Для доступа может потребоваться логин «admin» и пароль, хранящийся в переменной **mailion_internal_web_auth.password**.

4.2.4 Alertmanager

Alertmanager – система алертинга. Веб-интерфейс Alertmanager доступен по адресу `http://alertmanager.<infrastructure_inventory_hostname>/`.

Где `<infrastructure_inventory_hostname>` - FQDN хоста группы **ucs_infrastructure** (см. Рисунок 7).

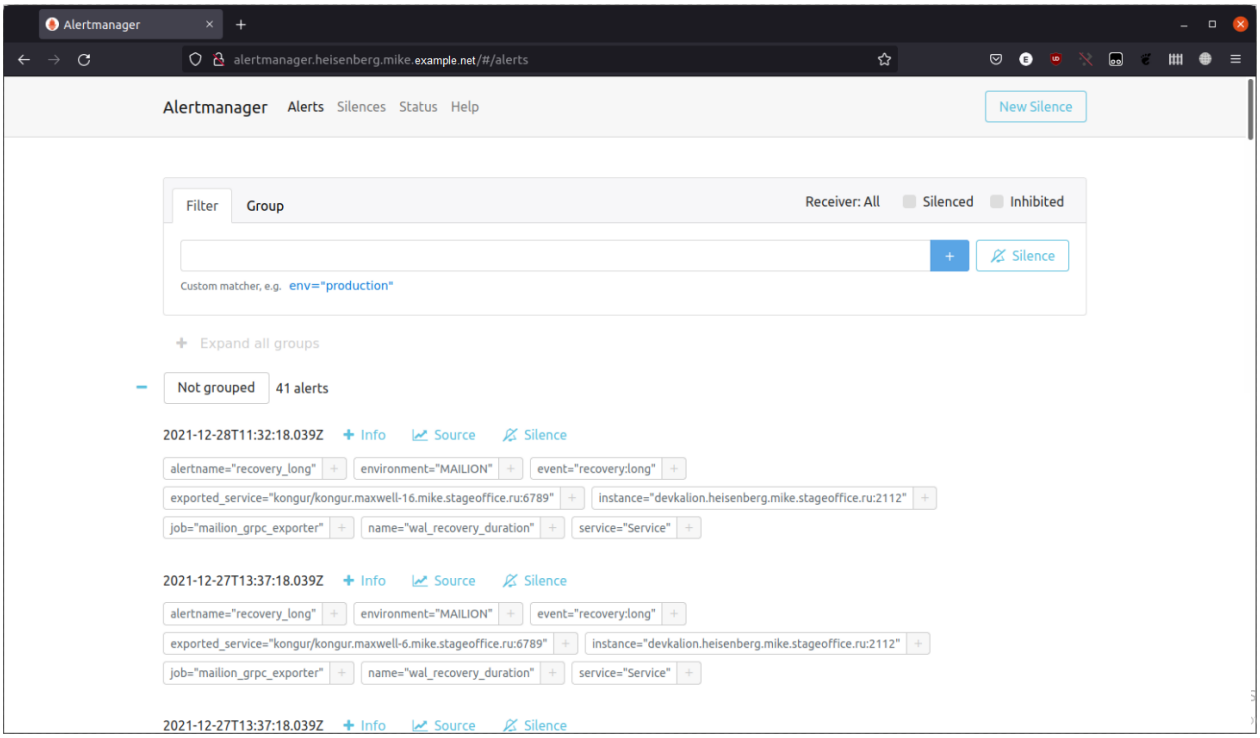


Рисунок 7 – Веб-интерфейс Alertmanager

Для доступа может потребоваться логин «admin» и пароль, хранящийся в переменной **mailion_internal_web_auth.password.**

4.2.5 Grafana

Grafana – система отображения метрик. Веб-интерфейс Grafana доступен по адресу `http://grafana.<infrastructure_inventory_hostname>`.

Где `<infrastructure_inventory_hostname>` - FQDN хоста группы **ucs_infrastructure** (см. Рисунок 8).

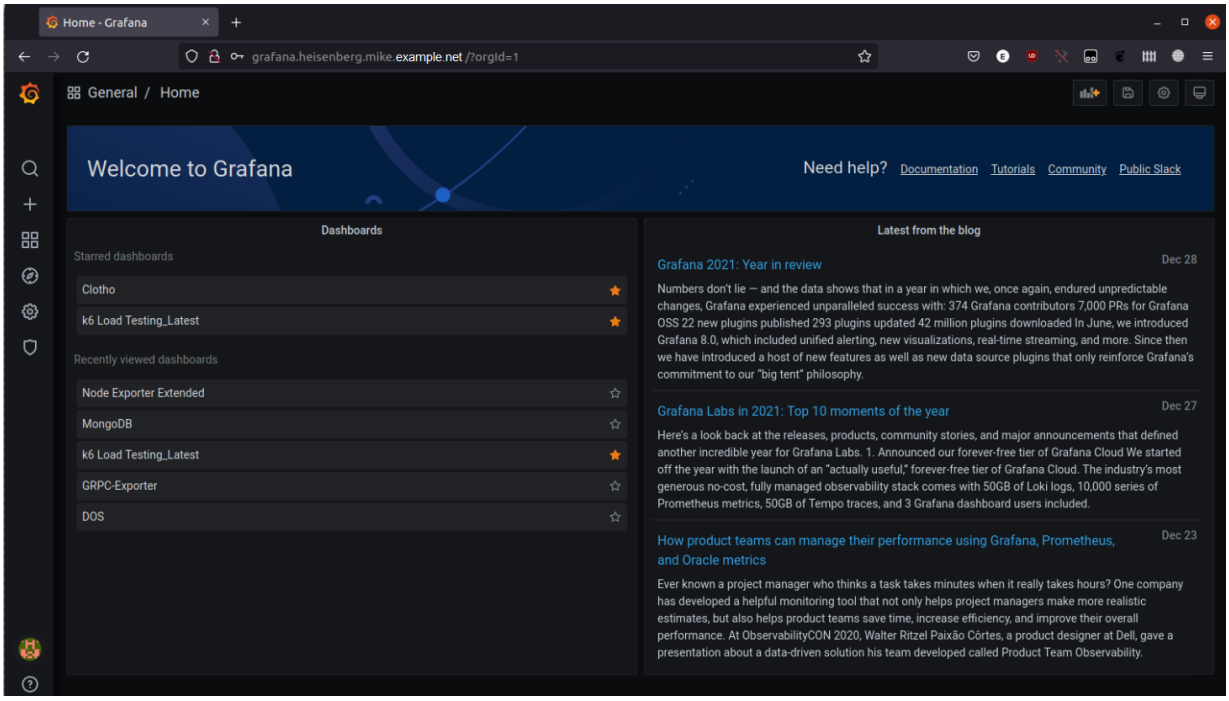


Рисунок 8 – Веб-интерфейс Grafana

Для доступа может потребоваться логин «admin» и пароль, хранящийся в переменной **grafana_admin_password**.

4.3 Настройка взаимодействия со службой каталогов

Для настройки интеграции с одним из каталогов (Microsoft Active Directory, FreeIPA, ALD Pro, РЕД АДМ и Samba DC) до инсталляции необходимо в соответствующем словаре указать уникальный ключ, в котором будут храниться параметры интеграции. Ключ можно сгенерировать с помощью команды «**pwgen 25 1**».

Настройки интеграции необходимо прописать в файле **group_vars/ucs_setup/main.yml/**. Пользователь, который прописывается в секции **bind_user** в данном конфигурационном файле, должен иметь права доступа на чтение к дереву Microsoft Active Directory, FreeIPA, ALD Pro и Samba DC.

Корректно заполненные параметры приведены ниже.

```
integrations:
 microsoft:
   "IS7YluhZ318G7Sm89SkkfZbO":
     ads:
       base_dn: "dc=example,dc=net"
       bind_user: "example\\aduser"
```

```
      bind_password: "adUserPassword"
      name: "AD"
      servers:
        - endpoint: "dc.example.net:636"
          tls:
            ca_filename: "ca_example.net.pem"
            cert_file: ""
            key_file: ""
          use_tls: true
      use_dc: false
    exchanges:
      exchange_version: "Exchange2013_SP1"
      ca_filename: ""
freeipa:
  "zuif6jeifiQueey5ahWatoo0o":
    dcs:
      base_dn: "dc=ipa-example,dc=net"
      bind_user: "uid=admin,cn=users,cn=accounts,dc=ipa-example,dc=net"
      bind_password: "adminPassword"
      name: "FreeIPA"
      servers:
        - endpoint: "freeipa.ipa-example.net:389"
          tls:
            ca_filename: ""
            cert_file: ""
            key_file: ""
          use_tls: false
      use_dc: false
samba_dc:
  "PeZh0WisXah5thooWhoo9bgG":
    smb:
      base_dn: "DC=samba-dc-test,DC=example,DC=com"
      bind_user: "Administrator"
      bind_password: "ahTh6uu7sah4so1C"
      name: "SAMBA_DC"
      servers:
        - endpoint: "samba-dc-test.example.com:389"
          tls:
            ca_filename: ""
            cert_file: ""
            key_file: ""
          use_tls: false
      use_dc: false
aldpro:
```

```
   "ALDh9ZisXah5thooWhoo9bgZ":
     ald:
       base_dn: "DC=domain,DC=test"
       bind_user: "admin"
       bind_password: "ahTh6uu7sah4so1C"
       name: "ALDPRO"
       servers:
         - endpoint: "aldpro-test.example.com:389"
           tls:
             ca_filename: ""
             cert_file: ""
             key_file: ""
           use_tls: false
       use_dc: false
```

Для включения интеграции с Microsoft Active Directory необходимо указать в групповых переменных:

```
mailion_integrations:
 microsoft: true
```

Для включения интеграции с FreeIPA необходимо указать в групповых переменных:

```
mailion_integrations:
 freeipa: true
```

Для включения интеграции с ALD Pro необходимо указать в групповых переменных:

```
mailion_integrations:
 aldpro: true
```

Для включения интеграции с Samba DC необходимо указать в групповых переменных:

```
mailion_integrations:
 samba_dc: true
```

Поддержка каталога РЕД АДМ осуществляется без заполнения параметров в конфигурационном файле.

В настройках для Microsoft Active Directory добавлена возможность конфигурировать используемую версию Exchange. Для этого используется переменная **exchange_version**. Она будет влиять на поддерживаемую версию Exchange, с которой идут запросы в EWS API. Данная переменная находится в разделе **exchanges**.

```
microsoft:
   ..................
   ads:
   ..................
   exchanges:
```

```
      ...................
      exchange_version: "Exchange2013_SP1"
```

Доступны следующие варинаты:

– "Exchange2010";

– "Exchange2010_SP1";

– "Exchange2010_SP2";

– "Exchange2013";

– "Exchange2013_SP1".

Если переменная не задана, то по умолчанию будет использовано значение "Exchange2013_SP1".

В настройках exchange присутствует поле **tls_min_version**. Оно содержит минимальную приемлемую версию TLS для работы с сервисами.

Данная настройка является обязательной для установки (значение по умолчанию не задано), без нее сервисы работать не будут.

Расположение переменной в файле конфигурации:

```
integrations:
  ...................
  microsoft:
    ...................
    exchanges:
      ...................
      servers:
          tls_min_version: "..."
```

В настройках exchange обязательным является поле **ca_filename**. Оно содержит имя файла сертификата, который необходимо скопировать в папку `~/install_mailion/certificates/` перед инсталляцией .

Расположение переменной **ca_filename** в файле конфигурации:

```
 microsoft:
    ...................
    ads:
    ...................
    exchanges:
      ...................
      ca_filename: ""
```

4.4 Настройка антивирусного программного обеспечения

В ПО «Mailion» **Rspamd** поддерживает несколько сторонних антивирусных модулей, в том числе KSE (Kaspersky). Настройка данного модуля осуществляется через переменные роли **Rspamd**. Подробное описание этих ролей приведено в таблицах 45 и 46.

Таблица 45 – Настройка Rspamd role vars

Параметр	Пример заполнения	Описание
rspamd:		
kse_use_https:	false	Использование **https** для подключения к серверам Касперского
kse_endpoints:	[]	Адреса серверов Касперского для обновления сигнатур (Обязательно наличие инсталляции KSE внутри компании)
kse_timeout:	"5.0"	Максимальный период времени для сканирования объекта
kse_scan_mime_parts:	true	Включение сканирования вложений
kse_use_files:	false	Отключение file mode в пользу TCP Stream. Не рекомендуется менять значение на true, режим file mode используется только для случаев наличия быстрой **tmpfs**
kse_max_size:	2048000	Максимальный размер файла для сканирования

Включение модуля антивирусной защиты Kaspersky осуществляется через групповые переменные инсталлятора ПО «Mailion», при наличии установленного в компании Сервера управления «Касперский антивирус».

Таблица 46 – Настройка Rspamd role vars

Параметр	Пример заполнения	Описание
rspamd:		
kse_enabled	true	Включение модуля Касперский для **rsmapd**
kse_endpoints:	"kaspersky.example.net:8085"	Список серверов управления антивирусной защитой Касперский

Продукт Kaspersky Scan Engine не является частью поставки ПО «Mailion».

4.5 Настройка сервиса imap

Для корректной работы сервиса imap необходимо убедиться, что файл конфигурации `/srv/docker/imap/conf/config.json` содержит следующие параметры:

```
{
...
"beef_client_cache": {"disable": true},
"tag_object_cache": {"disable": true},
...
"disable_audit": true,
...
}
```

4.6 Настройка сервиса Vault

Сервис хранения ключей Vault поддерживает многосерверный режим для обеспечения высокой доступности. Этот режим защищает от сбоев в работе за счет запуска нескольких серверов хранилища. Режим высокой доступности включается автоматически при использовании хранилища данных, которое его поддерживает.

Vault работает в такой схеме, когда все инстансы кластера развернуты и работоспособны, при этом только один инстанс активен. Он принимает запросы на чтение/запись, остальные инстансы остаются в режиме ожидания и перенаправляют все запросы на активный инстанс. Если активный инстанс выходит из строя, кластер сам выбирает новый активный хост, и система продолжает работать.

Все данные (секреты) автоматически синхронизируются и хранятся на всех трех инстансах. Количество инстансов Vault в кластере должно быть нечетным.

4.6.1 Установка сервиса Vault

Необходимо установить Vault на хосты vault1, vault2, vault3, при этом сам Vault не запускать и не распечатывать.

Для этого необходимо установить и запустить первый инстанс Vault.

Установка Vault сервера **осуществляется до установки остальных компонент** ПО «Mailion» один раз, в дальнейшем не нужно выполнять установку Vault, если она уже была выполнена или если нет рекомендаций по переустановке.

4.6.1.1 Этапы установки

1. Подготовить DNS-запись, по которой будет происходить обращение к сервису Vault.
2. Убедиться в доступности портов 8200, 8201, или других, если планируется их использовать на машине, которая будет предназначена для развертывания Vault.
3. Необходимо подготовить 3 файла для корректной работы сервиса Vault. Все файлы должны быть выпущены на доменное имя, подготовленное в предыдущем пункте, либо должны поддерживать Wildcard SSL сертификат, в который входит доменное имя из предыдущего пункта.

   Пример: если доменное имя **vault.example.ru**, то сертификаты должны быть выпущены либо на домен **vault.example.ru**, либо на ***.example.ru**. В последнем случае допустимо использовать сертификаты, уже подготовленные для инсталляции Mailion (см. раздел Размещение ssl-сертификатов):

   - CA сертификат, подписанный доверенным удостоверяющим центром (необходимо использовать для корректной работы);
   - сертификат сервера, подписанный подготовленным в предыдущем пункте приватным ключом CA;
   - приватный ключ для сертификата из предыдущего пункта.

4. Подготовленные ранее сертификаты необходимо расположить в директории коллекции **nct.certs/roles/tls_certs/files/**. Данная директория будет создана после распаковки установщика.
5. Необходимо обновить файл **inventory** и указать в нем созданные файлы. Пример секции в конфигурационном файле:

```
vault_tls:
  enabled: true
  certs:
    ca_filename: "<Имя CA файла>"
    cert_filename: "<Имя файла сертификата сервера>"
    key_filename: "<Имя файла ключа сервера>"
```

6. Выполнить команду установки:

```
install-mailion playbooks/mailion/vault.yml --tags=mln_vault -i <Путь к файлу inventory>
```

7. Необходимо распаковать сервис Vault. Для этого перейти в веб-браузере по адресу Vault сервиса с указанием порта и схемы (пример: https://vault.example.ru:8200). На странице будет предложено задать несколько параметров:

   - Key shares – количество ключей, которое будет сгенерировано;

- Key threshold - количество ключей из сгенерированных, которое понадобиться для распаковки Vault. Не может быть больше, чем Key shares.

8. Задать значения и инициализировать сервис. Будет предложено сохранить сгенерированные значения ключей для распаковки и токен для root-доступа на сервер в файл.

> ⚠ Необходимо обязательно сохранить файл! В случае рестарта сервиса Vault без ключей для распаковки его будет невозможно восстановить, так как все данные будут зашифрованы. Также для настройки понадобится root token, его можно будет перевыпустить, используя ключи для распаковки.

9. Ввести сохраненные ключи для распаковки, пока сервер не будет распакован полностью.

4.6.1.2 Настройка Vault AppRole и доступа к кластеру для приложений

Для настройки Vault AppRole и доступа к кластеру необходимо запустить команду:

```
install-mailion playbooks/mailion/vault.yml --tags=mln_vault_init -i <Путь к файлу inventory> -e vault_init_address=<Полный адрес до Vault сервера вместе с портом и схемой> -e vault_init_token=<Root токен, сохраненный на этапе инициализации Vault сервера>
```

Данная команда создает AppRole и необходимые политики доступа на Vault сервере, также она инициализирует пустой секрет по нужному пути.

В выводе данной команды будет указан токен **APP_ROLE_TOKEN** для доступа на Vault сервер для приложений ПО «Mailion», который нужно сохранить. Срок действия данного токена - 1 год. Для перевыпуска токена можно запустить команду еще раз.

4.6.1.3 Инициализация секретов

Чтобы создать список секретов необходимо выполнить следующие действия:

1. Проанализировать файл inventory и сохранить результат в файл с помощью команды:

```
grep -rni 'vault:.*' <Путь к директории с inventory файлами> | grep -o 'vault:.*' | sed 's/vault://g' | tr -d '\"' |  sort -h | uniq > first.txt
```

2. Проанализировать файл inventory на предмет других секретов с помощью команды:

```
grep -rnio 'vault_secrets\[.*\]' <Путь к директории с inventory файлами>  | grep -o '\[.*\]' | tr -d "[,']" | sort -h | uniq > second.txt
```

3. Сформировать финальный список секретов, которые необходимо внести в Vault. Для этого выполнить команду:

```
cat first.txt second.txt | sort -h | uniq > final_secret_list.txt
```

Сформированный файл будет содержать список секретов, которые необходимо внести в Vault. Значения для секретов заполняются самостоятельно.

4. Также необходимо проанализировать inventory файл на наличие открытых паролей, указанных в открытом виде. В случае наличия таковых, значения для них можно поменять на маскированные значения, и добавлять свои секреты в Vault. Подробная информация приведена в приложении Настройки inventory файла для работы с Vault).
5. В веб-браузере зайти на сервис Vault, используя полный адрес с портом.
6. Авторизоваться, используя root токен, полученный на этапе инициализации сервиса Vault.
7. Перейти во вкладку Secrets, в списке секретов перейти на **mailion** и далее на **installation**. Путь для секретов **mailion/installation**.
8. Нажать кнопку **Create new version** и заполнить в соответствии с полученными ранее секретами.
9. Создать секрет **vault_token** в качестве значения, указать токен для доступа приложений, полученный ранее на этапе настройки Vault AppRole (см. Настройка Vault AppRole и доступа к кластеру для приложений).
10. Для удобства можно заполнить в виде файла в формате JSON. Для этого есть специальная кнопка JSON, нужно ее включить.

4.6.1.4 Настройка .ansible.cfg для доступа к развернутому Vault серверу

Для настройки необходимо отредактировать файл **~/.ansible.cfg** на той машине, с которой планируется запускать установку ПО «Mailion», добавив следующую секцию:

```
[hashi_vault_collection]
token_path = /Users/user/projects/secrets/
token_file = vault.token # Внутри файла необходимо указать токен для приложений, полученный ранее на этапе настройки Vault AppRole
url = https://vault.server.company:8200 # URL, где будет доступен hosted vault - в случае с HA vault - достаточно одного из инстансов
```

4.6.1.5 Подготовка inventory файла

Для дальнейшей установки ПО «Mailion» необходимо подготовить inventory файл. При использовании Vault необходимо использовать `main.yml.hosted_vault`. Данный файл конфигурациии достаточно хорошо документирован, описание использования каждой секции подробно описано в комментариях.

Пример:

```
## ANSIBLE configuration
## remote SSH user with correct permissions
ansible_user: "root"
################################################################################
###################
# данный файл содержит актуальный пример настроек переменных
# в случае установки Mailion вместе с hosted vault
################################################################################
#########

################################################################################
#########
# данная секция обеспечивает
# интеграцию ansible с hosted vaut сервером
# для интеграции
# необходимо включить переменные:
# use_hashi_vault_secrets,
# use_hashi_vault_ad_secrets
.............................
```

4.6.2 Установка на другие хосты

Установка на другие хосты (vault2, vault3) осуществляется стандартным способом скачивания Vault дистрибутива с docker-контейнера, на котором запущен первый vault инстанс:

```
ssh infra # зайти на машину, на которой запущен докер контейнер с первым vault
инстансом
sudo docker cp vault:/usr/local/bin/vault /tmp/vault
sudo chmod a+r /tmp/vault
exit
ssh vault2
scp infra:/tmp/vault vault
sudo mv /tmp/vault /usr/local/bin/
./vault --version
```

4.6.3 Создание доменных имен

Опционально: для каждого из хостов, на DNS-серверах разворачиваемой инфраструктуры создать доменные имена:

− vault1.example.com;

− vault2.example.com;

− vault3.example.com.

Для этого перед установкой ПО «Mailion» необходимо добавить в файл (пример: ./папка_инсталяции/group_vars/ucs_Имя_Стенда/main.yml) следующие опции:

```
- type: "transparent"
    zone: "vault1.example.com"
    local_data:
      - domain: "vault1.example.com"
        type: "A"
        ip: "192.168.0.1"
  - type: "transparent"
    zone: "vault2.example.com"
    local_data:
      - domain: "vault2.example.com"
        type: "A"
        ip: "192.168.0.2"
  - type: "transparent"
    zone: "vault3.example.com"
    local_data:
      - domain: "vault3.example.com"
        type: "A"
        ip: "192.168.0.3"
```

На серверах с Vault рекомендуется добавить в файле **/etc/hosts** следующие записи:

```
192.168.0.1 vault1.example.com
192.168.0.2 vault2.example.com
192.168.0.3 vault3.example.com
```

Секции IP-адресов необходимо установить на соответствующие устанавливаемой конфигурации.

4.6.4 Генерация CA сертификата

Необходимо выпустить Wildcard SSL сертификат под домен для инсталляции ПО «Mailion» в аккредитованном центре сертификации. Либо использовать уже имеющийся

Wildcard SSL сертификат аккредитованного CA и формировать доменные имена третьего уровня исходя из имени домена, на который он был выдан.

В результате должны получиться три файла:

− server.crt;

− server.key;

− ca.pem.

4.6.5 Создание сертификатов для каждого инстанса

Для работы Vault необходимы сертификаты в формате PEM. Ключи могут уже быть в формате PEM, но им просто будет присвоено имя .crt или .key.

Если они начинаются с -----BEGIN и есть возможность прочитать их в текстовом редакторе (они используют base64, который читается в ASCII, а не в двоичном формате), то они находятся в формате PEM.

Если файлы в двоичном формате:

```
#для server.crt необходимо использовать:
openssl x509 -inform DER -outform PEM -in server.crt -out server_crt.pem

#для server.key необходимо использовать:
openssl rsa -inform DER -outform PEM -in server.key -out server_key.pem
```

Чтобы настроить параметр **tls_cert_file** в секции **listener** на использование сертификата CA, объедините основной сертификат (server.crt) и сертификат CA (ca.pem) в одном файле server.pem:

```
cat server.crt ca.pem > server.pem
```

В результате файл server.pem должен содержать сертификат и цепочку корневого сертификата:

```
cat server.pem
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

После этого необходимо скопировать сертификаты, созданные на предыдущем шаге в пути установки Vault и изменить владельцев и права доступа:

```
cp ./server_key.pem ./server.pem ./ca.pem /opt/vault/tls
chown root:root /opt/vault/tls/ca.pem
chown root:root /opt/vault/tls/server.pem
chown root:vault /opt/vault/tls/server_key.pem
chmod 0644 /opt/vault/tls/ca.pem
chmod 0644 /opt/vault/tls/server.pem
chmod 0644 /opt/vault/tls/server_key.pem
```

Для управления верификацией используется переменная

othrys_insecure_skip_verify: true

Значение **false** включает верификацию недоверенных сертификатов, значение **true** игнорирует ее

4.6.6 Настройка конфигурационного файла Vault для каждого инстанса

Для каждого из хостов, где установлен Vault необходимо создать один и тоже конфигурационный файл с разницей только в том, что необходимо указать доменное имя хоста и IP адрес, по которому Vault будет доступен.

Во всех секциях, где необходимо указать TLS сертификаты, нужно указать сертификат/ключ, а также CA сертификат, созданные на предыдущем шаге.

Для первого инстанса Vault, запущенного, как docker контейнер, предварительно рекомендуется сделать резервную копию директории **/srv/docker/vault**.

Далее необходимо отредактировать конфигурационный файл по пути **/srd/docker/vault/conf/config.hcl** способом, аналогичным другим инстансам. После того, как он будет отредактирован, необходимо перезапустить контейнер с помощью команды **sudo docker restart vault**.

```
ssh infra # машина в кластере mailion, на которой запущен первый vault инстанс
cat << HERE > /srv/docker/vault/config/vault.hcl
cluster_addr = "https://192.168.0.1:8201
api_addr = "https://192.168.0.1:8200
disable_mlock = true
ui = true

 listener "tcp" {
  address = "0.0.0.0:8200"
  tls_client_ca_file = "/opt/vault/tls/ca.pem"
  tls_cert_file = "/opt/vault/tls/server.pem"
  tls_key_file = "/opt/vault/tls/server_key.pem"
 }
```

```
# секция raft содержит в себе ссылки на _все_ инстансы vault
# доступные в кластере (включая инстанс, который установлен на данном хосте)
 storage "raft" {
  path = "/opt/vault/data"
  node_id = "vault1.example.com"

 retry_join {
  leader_tls_servername = "vault1.example.com"
  leader_api_addr = "https://192.168.0.1:8200
  leader_ca_cert_file = "/opt/vault/tls/ca.pem"
  leader_client_cert_file = "/opt/vault/tls/server.pem"
  leader_client_key_file = "/opt/vault/tls/server_key.pem"
 }
 retry_join {
  leader_tls_servername = "vault2.example.com"
  leader_api_addr = "https://192.168.0.2:8200
  leader_ca_cert_file = "/opt/vault/tls/ca.pem"
  leader_client_cert_file = "/opt/vault/tls/server.pem"
  leader_client_key_file = "/opt/vault/tls/server_key.pem"
 }
 retry_join {
  leader_tls_servername = "vault3.example.com"
  leader_api_addr = "https://192.168.0.3:8200
  leader_ca_cert_file = "/opt/vault/tls/ca.pem"
  leader_client_cert_file = "/opt/vault/tls/server.pem"
  leader_client_key_file = "/opt/vault/tls/server_key.pem"
 }
}
HERE
ssh vault2
[root@vault1] cat << HERE > /etc/vault.d/vault.hcl
cluster_addr  = "https://192.168.0.1:8201
api_addr      = "https://192.168.0.1:8200
disable_mlock = true

ui = true

listener "tcp" {
  address             = "0.0.0.0:8200"
  tls_client_ca_file  = "/opt/vault/tls/ca.pem"
  tls_cert_file       = "/opt/vault/tls/server.pem"
  tls_key_file        = "/opt/vault/tls/server_key.pem"
}
```

```
# секция raft содержит в себе ссылки на _все_ инстансы vault
# доступные в кластере (включая инстанс, который установлен на данном хосте)
storage "raft" {
  path    = "/opt/vault/data"
  node_id = "vault2.example.com"

  retry_join {
    leader_tls_servername   = "vault1.example.com"
    leader_api_addr         = "https://192.168.0.1:8200
    leader_ca_cert_file     = "/opt/vault/tls/ca.pem"
    leader_client_cert_file = "/opt/vault/tls/server.pem"
    leader_client_key_file  = "/opt/vault/tls/server_key.pem"
  }
  retry_join {
    leader_tls_servername   = "vault2.example.com"
    leader_api_addr         = "https://192.168.0.2:8200
    leader_ca_cert_file     = "/opt/vault/tls/ca.pem"
    leader_client_cert_file = "/opt/vault/tls/server.pem"
    leader_client_key_file  = "/opt/vault/tls/server_key.pem"
  }
  retry_join {
    leader_tls_servername   = "vault3.example.com"
    leader_api_addr         = "https://192.168.0.3:8200
    leader_ca_cert_file     = "/opt/vault/tls/ca.pem"
    leader_client_cert_file = "/opt/vault/tls/server.pem"
    leader_client_key_file  = "/opt/vault/tls/server_key.pem"
  }
}
HERE
```

4.6.7 Рестарт, распечатка первого инстанса Vault

Когда конфигурационные файлы готовы, необходимо на первый узел (vault1) и запустить сервис Vault с помощью команды:

```
ssh infra
docker restart vault
#  зайти в web интерфейс и распечатать
```

4.6.8 Запуск и распечатка остальных инстансов Vault

Для всех остальных нод кластера необходимо запустить и распечатать сервис Vault.

Инициализация для данных инстансов не требуется.

```
ssh vault2
sudo apt-get install screen
screen # мы будем использовать screen для запуска vault в режиме сервиса
# вы также можете создать systemd unit для этого и запускать vault сервис через systemd
# следуйте руководству вашего Linux дистрибутива о том, как создавать новые systemd сервисы

sudo vault server
^D # выход из screen сессии
vault operator unseal
```

Для распечатки необходимо следовать процедуре, описанной в разделе Рестарт, распечатка первого инстанса Vault. Использовать те же ключи распечатки, которые использовали для распечатки первого инстанса. Пример **systemd** конфигурации для сервиса Vautl. Выполняется с помощью команды **nano /etc/systemd/system/vault.service**.

```
[Unit]
Description=a tool for managing secrets
Documentation=https://vaultproject.io/docs/
After=network.target
ConditionFileNotEmpty=/etc/vault.d/vault.hcl

[Service]
User=vault
Group=vault
ExecStart=/usr/local/bin/vault server -config=/etc/vault.d/vault.hcl
ExecReload=/usr/local/bin/kill --signal HUP $MAINPID
CapabilityBoundingSet=CAP_SYSLOG CAP_IPC_LOCK
Capabilities=CAP_IPC_LOCK+ep
SecureBits=keep-caps
NoNewPrivileges=yes
KillSignal=SIGINT

[Install]
WantedBy=multi-user.target
```

Запуск через system:

```
systemctl daemon-reload
systemctl enable --now vault.service
systemctl start  vault.service
systemctl status vault.service
```

4.6.9 Верификация работы кластера

На данном этапе сформирован кластер из трех нод. Для верификации необходимо вернуться на вторую ноду и проверить состояние кластера.

Необходимо авторизоваться с токеном, который был получен ранее:

```
ssh vault2
vault login
Token (will be hidden):
Success! You are now authenticated. The token information displayed below
is already stored in the token helper. You do NOT need to run "vault login"
again. Future Vault requests will automatically use this token.
```

Затем проверить статус хранилища:

```
[root@vault1]# vault operator raft list-peers
Node                    Address             State       Voter
----                    -------             -----       -----
vault1.example.com    192.168.0.1:8201    leader      true
vault2.example.com    192.168.0.2:8201    follower    true
vault3.example.com    192.168.0.3:8201    follower    true
```

В выводе показаны три сервера, один из которых имеет статус ведущего (`leader`), а два других — ведомые (`follower`).

4.7 Настройка аудита событий в формате CEF

Чтобы настроить регистрацию событий пользователя для аудита в формате CEF, необходимо внести в конфигурационный файл (по умолчанию: `~/install_mailion/group_vars/ucs_setup/main.yml`) следующие изменения:

1. Включить функцию регистрации событий в формате CEF:

```
mailion_global_cef_enabled: true
```

2. Указать для сервиса **homeros** уровень журналирования TRACE, добавив следующий параметр:

```
homeros_log_level: "trace"
```

Если подключение настраивается на внешнюю SIEM-систему, то выполнять пункт 2 не требуется.

3. Указать адрес и порт внешней SIEM-системы с помощью следующего параметра:

```
homeros_cef_siem_endpoints:
   - "<адрес>:<порт>"
```

Если необходимо, чтобы события сохранялись в журнал сервиса **homeros**, следует указать адрес и порт, который будет настроен на контейнер **syslog_ng** (по умолчанию: 172.17.0.1:514), при этом события для аудита будут записываться на машину группы хостов **ucs_infrastructure** в каталог `~/srv/logs/ucs_setup/homeros/homeros.log`. Пример:

```
homeros_cef_siem_endpoints:
   - "172.17.0.1:514"
```

Если установка была сделана без этих изменений, то для применения настроек необходимо выполнить следующую команду из каталога `~/install_mailion`:

```
ansible-playbook -i hosts.yml playbooks/mailion.yml \
--tags homeros,house,cox \
--diff
```

5 КАТАСТРОФОУСТОЙЧИВОСТЬ

5.1 Предварительная настройка (до репликации)

1. Необходимо установить два экземпляра (две инсталляции) Mailion в двух ЦОД. Версии сервисов должны быть одинаковыми в обоих ЦОД. В данном документе подразумевается, что в контексте катастрофоустойчивости ЦОД1 изначально является основным кластером (Active), а ЦОД2 — резервным (Standby).
2. Для обеспечения катастрофоустойчивости требуется БД MongoDB версии 6.0.14-32. Прежде всего, необходимо убедиться, что задан параметр `mongodb_image_tag: "6.0.14-32"`. Затем для проверки версии MongoDB, необходимо зайти на любую из нод с базами данных и выполнить команду `docker ps -a | grep mongodb_generic`. Если установлена более ранняя версия MongoDB, необходимо обновить ее (см. раздел Обновление Mongodb до версии 6.0).
3. В локальный кэширующий DNS-сервер (**unbound**) добавить ноды других кластеров или любым другим способом обеспечить разрешение имен (DNS) между кластерами. Список нод, которые нужно добавить:
   - где развернуты сервисы поиска **mailbek-search** и **dirbek** — группа **ucs_search**;
   - где развернут сервис **dispersed_object_store** — группа **dispersed_object_store**;
   - для ноды, на которой будет запущен репликатор MongoShake, должны быть доступны ноды, где развернута MongoDB — **ucs_mongodb**.

MongoShake — сторонний инструмент репликации MongoDB.

4. В `group_vars` для ЦОД1 и ЦОД2 добавить переменную **dorofej_region_id** и в обоих ЦОД задать для нее ОДИНАКОВЫЕ значения в том формате, в котором этот параметр хранится в **MongoDB**. Пример:

```
dorofej_region_id: '0b680cec-36be-4ee5-8c92-b5c4a5ae9011'
```

В случае разных значений **region_id** в основном и резервным кластерах часть сервисов может работать некорректно, также могут возникать проблемы с пересылкой писем.

Если переменная не была добавлена на этапе установки или получилось так, что **region_id** имеет разные значения в основном и резервном кластере, то **ТОЛЬКО ПОСЛЕ НАСТРОЙКИ РЕПЛИКАЦИИ** необходимо:

- На всех хостах резервного кластера поменять в конфигурационных файлах **region_id**, например с помощью команды:

```
find /srv/docker/*/conf* -type f -exec sed -i s/<старый region_id>/<новый region_id>/g {} \;
```

- Перезапустить docker-демон, например командой:

```
sudo systemctl restart docker.service
```

5. **Задать единый SSID для cookie катастрофоустойчивости**.

Единый SSID нужен для внутренних автотестов и нагрузочного тестирования.

- При развертывании сервиса **house** задать в **group_vars** переменную:

```
house_cookie_name=<любое символьное значение, например SSID_UCS_DISASTER>
```

- После переключения пользователям, возможно, понадобится авторизоваться повторно.
- Если эта переменная имеет разные значения на этих двух ЦОД, то необходимо указать одинаковые значения для обоих ЦОД и заново развернуть сервис **house**. В случае, если репликация уже настроена (сервис **house** остановлен), то вручную поменять конфигурацию на виртуальной машине. Для этого:
  - Зайти на ВМ с сервисом **house**:

    ```
    ssh ucs-frontend-1.mailion-disaster-02.example.com
    ```
  - Проверить текущее значение:

    ```
    sudo grep -Rni 'SSID_' /srv/docker/house/conf/
    ```
  - Поменять в файле значение:

    ```
    sed -i 's/SSID_UCS_DISASTER_02_MSK/SSID_UCS_DISASTER/g' /srv/docker/house/conf/conf.d/general_settings.conf
    ```
  - Если на ЦОД1 меняется конфигурация, и он является основным, то необходимо перезапустить контейнер **house**. Если конфигурация меняется в резервном кластере (сервисы остановлены), то перезапускать **house** не требуется.
  - Повторить для всех машин, где запущен фронтенд **house**.

6. **Убедиться, что выключен скрипт, который чистит остановленные контейнеры**. По умолчанию в поставку входит скрипт, который чистит некоторые остановленные контейнеры. Это может приводить к удалению контейнеров с сервисами на резервном кластере. Для остановки этого скрипта необходимо при установке кластера задать параметр:

   ```
   docker_cleaner_enabled: false
   ```

7. **Проверить выполнение требований для развертывания сервисов поиска (mailbek_search и dirbek)**. При развертывании поиска конфигурационные файлы создаются автоматически. При установке сервисов поиска необходимо, чтобы для ЦОД1 и ЦОД2 были заданы определенные параметры.

- Для ЦОД1:

```
# Search services sync config
mailbek_sync_enabled: true
dirbek_sync_enabled: true
dirbek_service_mode: "dir_replic"
# mailbek_service_mode: "replic" #Replic by default

# This is used for sync section on dirbek and mailbek
ucs_replic:
  - "ucs-search-1.mailion-disaster-02.example.com"
  - "ucs-search-2.mailion-disaster-02.example.com"
  - "ucs-search-3.mailion-disaster-02.example.com"

mailbek_sync_tls_ca_file: "ucs-infra-1.mailion-disaster-02.example.com-main-
ca.pem"
mailbek_sync_tls_client_cert_file: "mailbek-search.ucs-search-{{ search_segment_id
}}.mailion-disaster-02.example.com-main-client.pem"
mailbek_sync_tls_key_file: "mailbek-search.ucs-search-
{{ search_segment_id }}.mailion-disaster-02.example.com-main-key.pem"
mailbek_sync_tls_server_cert_file: "mailbek-search.ucs-search-{{ search_segment_id
}}.mailion-disaster-02.example.com-main-server.pem"

dirbek_sync_tls_ca_file: "ucs-infra-1.mailion-disaster-02.example.com-main-ca.pem"
dirbek_sync_tls_client_cert_file: "dirbek.ucs-search-
{{ search_segment_id }}.mailion-disaster-02.example.com-main-client.pem"
dirbek_sync_tls_key_file: "dirbek.ucs-search-{{ search_segment_id }}.mailion-
disaster-02.example.com-main-key.pem"
dirbek_sync_tls_server_cert_file: "dirbek.ucs-search-
{{ search_segment_id }}.mailion-disaster-02.example.com-main-server.pem"
```

Блок **ucs_replic** содержит список хостов противоположного кластера (для ЦОД1 это список хостов из ЦОД2, а для ЦОД2 — соответственно из ЦОД1). Этот список нужен для того, чтобы в конфигурациях сервисов появились нужные секции для настройки репликации.

- Для ЦОД2:

```
# Search services sync config
mailbek_sync_enabled: true
dirbek_sync_enabled: true
dirbek_service_mode: "dir_replic"
# mailbek_service_mode: "replic" #Replic by default

tls_certs_dir: "/srv/tls/certs"
tls_keys_dir: "/srv/tls/keys"

# This is used for sync section on dirbek and mailbek
ucs_replic:
  - "ucs-search-1.mailion-disaster-01.example.com"
  - "ucs-search-2.mailion-disaster-01.example.com"
  - "ucs-search-3.mailion-disaster-01.example.com"

mailbek_sync_tls_ca_file: "ucs-infra-1.mailion-disaster-01.example.com-main-
ca.pem"
mailbek_sync_tls_client_cert_file: "mailbek-search.ucs-search-{{ search_segment_id
}}.mailion-disaster-01.example.com-main-client.pem"
mailbek_sync_tls_key_file: "mailbek-search.ucs-search-
{{ search_segment_id }}.mailion-disaster-01.example.com-main-key.pem"
mailbek_sync_tls_server_cert_file: "mailbek-search.ucs-search-{{ search_segment_id
}}.mailion-disaster-01.example.com-main-server.pem"

dirbek_sync_tls_ca_file: "ucs-infra-1.mailion-disaster-01.example.com-main-ca.pem"
dirbek_sync_tls_client_cert_file: "dirbek.ucs-search-
{{ search_segment_id }}.mailion-disaster-01.example.com-main-client.pem"
dirbek_sync_tls_key_file: "dirbek.ucs-search-{{ search_segment_id }}.mailion-
disaster-01.example.com-main-key.pem"
dirbek_sync_tls_server_cert_file: "dirbek.ucs-search-
{{ search_segment_id }}.mailion-disaster-01.example.com-main-server.pem"
```

5.2 Настройка репликации

Репликация реализуется следующим образом:

- MongoDB — с помощью MongoShake;
- DOS — поддерживается самим сервисом;
- поиск (mailbek, dirbek) — поддерживается самими сервисами.

Начальные условия:

- Две инсталляции Mailion 2.0: **disaster-01-msk** (основная, в ЦОД1) и **disaster-02-msk** (резервная, в ЦОД2).
- DNS-сервер настроен на ЦОД1: **disaster-msk.example.com**.
- Репликация не настроена, обе инсталляции работают независимо друг от друга.

1. **Скопировать сертификаты между ЦОД**

- Подготовить inventory-файл. Ниже приведен пример такого файла со списком всех необходимых сертификатов для копирования между ЦОД1 и ЦОД2.

```
 ---

all:
  children:
    ucs:
      children:

      ### Cluster one
        ucs_dos_cluster_one:
          hosts:
            ucs-obst-[1:4].mailion-disaster-01.example.com:
          vars:
            certs_copy_to:
              - "/srv/tls/certs/ucs-infra-1.mailion-disaster-02.example.com-main-
ca.pem"

        ucs_apps_cluster_one:
          hosts:
            ucs-apps-1.mailion-disaster-01.example.com:
          vars:
            certs_copy_from:
              - "/srv/tls/certs/sophokles.ucs-apps-1.mailion-disaster-
01.example.com-main-client.pem"
              - "/srv/tls/keys/sophokles.ucs-apps-1.mailion-disaster-
01.example.com-main-key.pem"

        ucs_mongodb_cluster_one:
          hosts:
            ucs-db-1.mailion-disaster-01.example.com:
          vars:
            certs_copy_from:
              - "/srv/tls/certs/merged_mongodb.ucs-db-1.mailion-disaster-
01.example.com-main-peer.pem"

        ucs_infrastructure_cluster_one:
          hosts:
            ucs-infra-1.mailion-disaster-01.example.com:
          vars:
            certs_copy_from:
              - "/srv/tls/certs/ucs-infra-1.mailion-disaster-01.example.com-main-
ca.pem"

        ucs_search_cluster_one:
          hosts:
            ucs-search-1.mailion-disaster-01.example.com:
              search_id: 1
            ucs-search-2.mailion-disaster-01.example.com:
              search_id: 2
            ucs-search-3.mailion-disaster-01.example.com:
              search_id: 3
          vars:
            certs_copy_from:
              - "/srv/tls/certs/mailbek-search.ucs-search-{{ search_id }}.mailion-
disaster-01.example.com-main-client.pem"
              - "/srv/tls/certs/dirbek.ucs-search-{{ search_id }}.mailion-
disaster-01.example.com-main-client.pem"
              - "/srv/tls/keys/mailbek-search.ucs-search-{{ search_id }}.mailion-
disaster-01.example.com-main-key.pem"
              - "/srv/tls/keys/dirbek.ucs-search-{{ search_id }}.mailion-disaster-
01.example.com-main-key.pem"
            certs_copy_to:
```

```
              - "/srv/tls/certs/ucs-infra-1.mailion-disaster-02.example.com-main-
ca.pem"
              - "/srv/tls/certs/mailbek-search.ucs-search-{{ search_id }}.mailion-
disaster-02.example.com-main-client.pem"
              - "/srv/tls/certs/dirbek.ucs-search-{{ search_id }}.mailion-
disaster-02.example.com-main-client.pem"
              - "/srv/tls/keys/mailbek-search.ucs-search-{{ search_id }}.mailion-
disaster-02.msk.example.com-main-key.pem"
              - "/srv/tls/keys/dirbek.ucs-search-{{ search_id }}.mailion-disaster-
02.msk.example.com-main-key.pem"

        mongoshake_cluster_one:
          hosts:
            ucs-infra-1.mailion-disaster-01.msk.example.com:
          vars:
            certs_copy_to:
              - "/srv/tls/certs/ucs-infra-1.mailion-disaster-01.msk.example.com-
main-ca.pem"
              - "/srv/tls/certs/merged_mongodb.ucs-db-1.mailion-disaster-
01.msk.example.com-main-peer.pem"
              - "/srv/tls/certs/sophokles.ucs-apps-1.mailion-disaster-
01.msk.example.com-main-client.pem"
              - "/srv/tls/keys/sophokles.ucs-apps-1.mailion-disaster-
01.msk.example.com-main-key.pem"
              - "/srv/tls/certs/ucs-infra-1.mailion-disaster-02.msk.example.com-
main-ca.pem"
              - "/srv/tls/certs/merged_mongodb.ucs-db-1.mailion-disaster-
02.msk.example.com-main-peer.pem"
              - "/srv/tls/certs/sophokles.ucs-apps-1.mailion-disaster-
02.msk.example.com-main-client.pem"
              - "/srv/tls/keys/sophokles.ucs-apps-1.mailion-disaster-
02.msk.example.com-main-key.pem"

      ### Cluster two
        ucs_dos_cluster_two:
          hosts:
            ucs-obst-[1:4].mailion-disaster-02.msk.example.com:
          vars:
            certs_copy_to:
              - "/srv/tls/certs/ucs-infra-1.mailion-disaster-01.msk.example.com-
main-ca.pem"

        ucs_apps_cluster_two:
          hosts:
            ucs-apps-1.mailion-disaster-02.msk.example.com:
          vars:
            certs_copy_from:
              - "/srv/tls/certs/sophokles.ucs-apps-1.mailion-disaster-
02.msk.example.com-main-client.pem"
              - "/srv/tls/keys/sophokles.ucs-apps-1.mailion-disaster-
02.msk.example.com-main-key.pem"

        ucs_mongodb_cluster_two:
          hosts:
            ucs-db-1.mailion-disaster-02.msk.example.com:
          vars:
            certs_copy_from:
                - "/srv/tls/certs/merged_mongodb.ucs-db-1.mailion-disaster-
02.msk.example.com-main-peer.pem"

        ucs_infrastructure_cluster_two:
          hosts:
            ucs-infra-1.mailion-disaster-02.msk.example.com:
          vars:
            certs_copy_from:
              - "/srv/tls/certs/ucs-infra-1.mailion-disaster-02.msk.example.com-
main-ca.pem"
```

```
        ucs_search_cluster_two:
          hosts:
            ucs-search-1.mailion-disaster-02.msk.example.com:
              search_id: 1
            ucs-search-2.mailion-disaster-02.msk.example.com:
              search_id: 2
            ucs-search-3.mailion-disaster-02.msk.example.com:
              search_id: 3
          vars:
            certs_copy_from:
              - "/srv/tls/certs/mailbek-search.ucs-search-{{ search_id }}.mailion-
disaster-02.msk.example.com-main-client.pem"
              - "/srv/tls/certs/dirbek.ucs-search-{{ search_id }}.mailion-
disaster-02.msk.example.com-main-client.pem"
              - "/srv/tls/keys/mailbek-search.ucs-search-{{ search_id }}.mailion-
disaster-02.msk.example.com-main-key.pem"
              - "/srv/tls/keys/dirbek.ucs-search-{{ search_id }}.mailion-disaster-
02.msk.example.com-main-key.pem"
            certs_copy_to:
              - "/srv/tls/certs/ucs-infra-1.mailion-disaster-01.msk.example.com-
main-ca.pem"
              - "/srv/tls/certs/mailbek-search.ucs-search-{{ search_id }}.mailion-
disaster-01.msk.example.com-main-client.pem"
              - "/srv/tls/certs/dirbek.ucs-search-{{ search_id }}.mailion-
disaster-01.msk.example.com-main-client.pem"
              - "/srv/tls/keys/mailbek-search.ucs-search-{{ search_id }}.mailion-
disaster-01.msk.example.com-main-key.pem"
              - "/srv/tls/keys/dirbek.ucs-search-{{ search_id }}.mailion-disaster-
01.msk.example.com-main-key.pem"


        mongoshake_cluster_two:
          hosts:
            ucs-infra-1.mailion-disaster-02.msk.example.com:
          vars:
            certs_copy_to:
              - "/srv/tls/certs/ucs-infra-1.mailion-disaster-01.msk.example.com-
main-ca.pem"
              - "/srv/tls/certs/merged_mongodb.ucs-db-1.mailion-disaster-
01.msk.example.com-main-peer.pem"
              - "/srv/tls/certs/sophokles.ucs-apps-1.mailion-disaster-
01.msk.example.com-main-client.pem"
              - "/srv/tls/keys/sophokles.ucs-apps-1.mailion-disaster-
01.msk.example.com-main-key.pem"
              - "/srv/tls/certs/ucs-infra-1.mailion-disaster-02.msk.example.com-
main-ca.pem"
              - "/srv/tls/certs/merged_mongodb.ucs-db-1.mailion-disaster-
02.msk.example.com-main-peer.pem"
              - "/srv/tls/certs/sophokles.ucs-apps-1.mailion-disaster-
02.msk.example.com-main-client.pem"
              - "/srv/tls/keys/sophokles.ucs-apps-1.mailion-disaster-
02.msk.example.com-main-key.pem"
      vars:
        local_cert_dir: "/tmp"
```

В этом файле для каждого хоста или группы хостов заданы следующие переменные:

– **certs_copy_from** — сертификаты с полным путем для копирования с данного хоста или группы хостов;

– **certs_copy_to** — сертификаты с полным путем для копирования на данный хост или группу хостов.

В обоих случаях сертификаты указываются **с полным путем к ним** (по которому они хранятся на хостах). Сертификаты копируются и выкладываются по полным указанным путям.

Параметр **local_cert_dir** — каталог на машине оператора, куда будут сохранены сертификаты. В конце этого значения **НЕ ДОЛЖНО БЫТЬ КОСОЙ ЧЕРТЫ '/'**.

- Выполнить команду:

```
ansible-playbook playbooks/mailion/disaster/copy_certs.yml \
        -e private_ansible_user=astra \
        -b \
        --private-key ~/.ssh/myoffice/ldap -i
inventory/ucs_msk_disaster_copy_certs.yml
```

2. **Создать резервные копии БД MongoDB в каждом ЦОД**

- На машине **infra** каждого ЦОД вручную запустить **bash**-скрипт для создания резервных копий (включен в поставку):

  `/srv/docker/mongodb/backup_scripts/mongodb_backup_generic.sh`

  ○ Сохранить резервные копии в любое удобное место.

В поставку также включена **cron**-задача, которая по умолчанию делает резервные копии MongoDB каждый день в 01:00 при условии, что во время установки Mailion она не была выключена и был установлен плейбук для резервного копирования.

3. **Создать резервную копию MongoDB в ЦОД1 для будущего восстановления в ЦОД2 перед запуском MongoShake**

Резервную копию в ЦОД1 можно создать так же, как описано в предыдущем разделе. MongoShake переносит данные из ЦОД1 в ЦОД2, но только те, которые есть в журнале операций в ЦОД1. Объем журнала операций MongoDB в ЦОД1 ограничен, и старые данные, которые хранятся в MongoDB ЦОД1 уже могут быть вытеснены из журнала новыми данными в результате ротации. Поэтому, если пропустить этот шаг, то данные будут реплицированы в ЦОД2, но гарантии, что там окажутся все данные из MongoDB ЦОД1, нет.

4. **Остановить сервисы в ЦОД2**

- Выполнить команду:

```
ansible-playbook \
    -i inventory/ucs_disaster02_msk.yml
playbooks/mailion/disaster/disaster_stop_services.yml  \
    --private-key ~/.ssh/myoffice/awx \
    -e private_ansible_user=astra
```

- На веб-странице Kunkka проверить, что остались запущенными **hydra**, **mailbek-search**, **dirbek**, **dispersed-object-store**, **mongodb**, сервисы **infra**, в том числе **house** для **infra** (не для фронтенда).

В сервисе Graphana есть панель мониторинга для репликации под названием **disaster recovery**. На ней отображаются только **dos**, **mongoshake** и **postfix**. Сервисы поиска не отображаются — в них пока не реализована поддержка отображения метрик репликации. В текущий момент мы ожидаемо должны видеть «n/a» на графиках, так как репликация еще не включена.

5. **Очистить БД MongoDB в ЦОД2**

Этот шаг необходим, чтобы на этапе запуска **MongoShake** не было конфликтов при записи данных из ЦОД1 в ЦОД2.

- Настроить подключение к мастер-узлу набора реплик (master replica set) через стандартный клиент **MongoDB** — **mongosh**, указав логин и пароль, которые использовались при установке и развертывании Mailion:

```
mongosh \
--shell \
--tlsCertificateKeyFile ./certs/merged_mongodb.ucs-db-1.mailion-disaster-
02.msk.example.com-main-peer.pem \
--tlsCAFile ./certs/ucs-infra-1.mailion-disaster-02.msk.example.com-main-ca.pem \
--tlsAllowInvalidHostnames \
--tls \
--verbose \
--username <логин> \
--password <пароль> \
--port 27017 \
--host mongodb.ucs-db-3.mailion-disaster-02.msk.example.com
```

- Выполнить команды:

```
show dbs
use admin;
db.getMongo().getDBNames().filter(n => !
['admin','local','config'].includes(n)).forEach(dname =>
db.getMongo().getDB(dname).dropDatabase());
show dbs
```

6. **Удалить из MongoDB в ЦОД1 данные MongoShake**

Этот шаг необходим для сброса контрольных точек с прошлых запусков, чтобы при работе MongoShake учитывался весь журнал операций с самого начала. Если это первая установка, и репликация через MongoShake еще ни разу не настраивалась, этот шаг можно пропустить.

- Использовать процедуру настройки подключения к мастер-узлу набора реплик (master replica set) MongoDB из предыдущего пункта, но заменить обозначение ЦОД2 на ЦОД1).
- Выполнить команды:

```
use mongoshake
db.dropDatabase('mongoshake')
```

7. **Восстановить MongoDB в ЦОД2 из созданной ранее резервной копии в ЦОД1, используя bash-скрипт**

- Зайти на машину **infra** в ЦОД1.

```
BACKUP_FILE="mongodb_dump_2024_08_15_1125.gz"
BACKUP_DIR="/srv/backups/mongodb/"
MONGODB_ROOT_PASSWORD="DB_PASS"

DB_LIST=(
  "achill" \
    "beef" \
    "clotho" \
    "dafnis" \
    "daidal" \
    "dorofej" \
    "ektor" \
    "erakles" \
    "eratosthenis" \
    "euripides" \
    "hog" \
    "homeros" \
    "kongur" \
    "kronos" \
    "marker" \
    "odusseus" \
    "perseus" \
    "sophokles" \
    "talaos" \
    "themis" \
    "theseus" \
    "thoth"
)

for db in ${DB_LIST[@]}; do
        echo $db;
        docker run -it --rm -v ${BACKUP_DIR}:/backups -
v /srv/tls/certs:/etc/pki/tls/certs \
        -e 'MONGO_CONN=mongodb://root:${MONGODB_ROOT_PASSWORD}@mongodb.ucs-db-
1.mailion-disaster-02.msk.example.com:27017,mongodb.ucs-db-2.mailion-disaster-
02.msk.example.com:27017,mongodb.ucs-db-3.mailion-disaster-
02.msk.example.com:27017/?
authSource=admin&replicaSet=ucs&tls=true&tlsCAFile=/etc/pki/tls/certs/ucs-infra-
1.mailion-disaster-02.msk.example.com-main-
ca.pem&tlsCertificateKeyFile=/etc/pki/tls/certs/merged_mongodb.ucs-db-1.mailion-
disaster-02.msk.example.com-main-peer.pem' \
        --name mongorestore hub.example.com/mongo:6.0.14-32 \
        sh -c "mongorestore --drop  --gzip \$MONGO_CONN --
authenticationDatabase=admin --nsInclude=\"$db.*\" --maintainInsertionOrder --
numParallelCollections=1 --numInsertionWorkersPerCollection=1 --objcheck --
convertLegacyIndexes --stopOnError  --archive=/backups/${BACKUP_FILE}"
done
```

- В скрипте подставить логин и пароль для MongoDB, которые использовались при установке Mailion. Помимо этого, можно поменять имя файла резервной копии и путь к нему, а также пути к файлам с сертификатами.

Необходимо оставить следующие параметры равными 1:
`--numParallelCollections=1`
`--numInsertionWorkersPerCollection=1`

В скрипте восстановление выполняется для каждой указанной базы в отдельности. Это нужно, чтобы избежать ошибок утилиты при восстановлении.

Пропускать этот шаг нельзя, иначе старые данные из БД в ЦОД1 не попадут в MongoDB ЦОД2 из-за возможной ротации журнала операций.

8. Запустить MongoShake

- Подготовить inventory-файл:

```
all:
  children:

### SECTION 1: grouping by Roles
    ## App: UCS
    ucs:
      children:
        ucs_mongodb_source:
          hosts:
            ucs-db-[1:3].mailion-disaster-01.msk.example.com:

        ucs_mongodb_target:
          hosts:
            ucs-db-[1:3].mailion-disaster-02.msk.example.com:

        ucs_infrastructure:
          hosts:
            ucs-infra-1.mailion-disaster-01.msk.example.com:

## SECTION 2: grouping by tier
    ucs_disaster_01_msk:
      # all servers should be listed
      hosts:
        ucs-infra-1.mailion-disaster-01.msk.example.com:
        ucs-db-[1:3].mailion-disaster-01.msk.example.com:
        ucs-db-[1:3].mailion-disaster-02.msk.example.com:
      vars:
        tier: "ucs_disaster_01_msk"
```

- Задать в **group_vars** ЦОД1 следующие переменные:

```
tls_certs_dir: "/srv/tls/certs"
tls_keys_dir: "/srv/tls/keys"

mongoshake_source_db_user: "root"
mongoshake_source_db_password: "SOURCE_MONGODB_ROOT_PASSWORD"
mongoshake_source_db_hosts: "{{ groups['ucs_mongodb_source'] }}"

mongoshake_target_db_user: "root"
mongoshake_target_db_password: "{{ mongodb_root_password }}"
mongoshake_target_db_hosts: "{{ groups['ucs_mongodb_target'] }}"

mongoshake_source_db_tls:
  tls_enabled: true
  root_ca_file: "ucs-infra-1.mailion-disaster-01.msk.example.com-main-ca.pem"
  merged_pem_file: "merged_mongodb.ucs-db-1.mailion-disaster-01.msk.example.com-main-peer.pem"
  client_cert: "sophokles.ucs-apps-1.mailion-disaster-01.msk.example.com-main-client.pem"
  client_key: "sophokles.ucs-apps-1.mailion-disaster-01.msk.example.com-main-key.pem"
mongoshake_target_db_tls:
  tls_enabled: true
  root_ca_file: "ucs-infra-1.mailion-disaster-02.msk.example.com-main-ca.pem"
  merged_pem_file: "merged_mongodb.ucs-db-1.mailion-disaster-02.msk.example.com-main-peer.pem"
  client_cert: "sophokles.ucs-apps-1.mailion-disaster-02.msk.example.com-main-client.pem"
  client_key: "sophokles.ucs-apps-1.mailion-disaster-02.msk.example.com-main-key.pem"

mongoshake_container_state: "stopped"
mongoshake_restart_policy: "no"

mongoshake_command:
  - "-conf"
  - "/apps/collector.conf"
  - "-verbose"
  - "2"
```

Для других параметров MongoShake в `group_vars` рекомендуется оставить значения по умолчанию.

Параметр **mongoshake_full_sync_create_index** должен иметь значение **background**. Это необходимо, чтобы при репликации данных утилита MongoShake сразу создавала индексы. По умолчанию этот параметр имеет значение **none**, и индексы в таком случае не создаются.

Параметр **mongoshake_full_sync_collection_exist_drop** должен иметь значение **false**. Если он будет иметь значение **true**, то при запуске репликации существующие коллекции в ЦОД2 будут удалены, и тогда в ЦОД2 останутся только те данные, которые указаны в журнале операций ЦОД1. **НИКОГДА не используйте значение true**.

Параметр **mongoshake_full_sync_executor_insert_on_dup_update** должен иметь значение **true**. Иначе возможна ситуация, когда MongoShake будет пытаться выполнить операцию **insert** над уже существующими записями, что вызовет аварийное завершение работы утилиты. При использовании значения **true** вместо операции **insert** будет выполняться операция **update**.

Для параметра **mongoshake_mongo_connect_mode** рекомендуется задавать значение **secondary_preferred**. При этом MongoShake в исходной БД MongoDB будет читать журнал операций вторичных узлов (через них не идет запись), а не мастер-узла набора реплик.

- Выполнить команду

```
ansible-playbook playbooks/mailion/disaster/mongoshake.yml \
        -e private_ansible_user=astra \
        --diff \
        --private-key ~/.ssh/myoffice/ldap -i
inventory/ucs_mongoshake_disaster_msk.yml
```

- Запустить контейнер MongoShake:
  - Зайти на машину ЦОД1, на которой создан контейнер MongoShake:

```
ssh ucs-infra-1.mailion-disaster-01.msk.example.com
```

  - Запустить скрипт:

```
/srv/docker/mongoshake/conf/start_mongoshake_vault.sh
```

  - Ввести пароль от исходной БД MongoDB в ЦОД1 и от целевой БД MongoDB в ЦОД2.
- Проверить ход репликации с помощью MongoShake:

```
http://ucs-infra-1.mailion-disaster-01.example.com:49100/repl
```

9. Выполнить сброс Redis в ЦОД2

После завершения установки и проверок в Redis остаются некоторые закешированные данные, которые могут помешать пользователям авторизоваться, когда ЦОД1 станет активным (сервис **minos**), и также могут мешать запуску сервисов (в журналах появляется сообщение об ошибке «*minos: right ids must not be empty*»).

- Запустить плейбук:

```
ansible-playbook -i inventory/ucs_disaster02_msk.yml
playbooks/mailion/disaster/redis_clear_cache.yml --extra-vars "ansible_user=astra"
-e redis_clear_cached_enabled=true -v
```

Для некоторых сервисов при установке Mailion кеш-кластеры Redis могут быть НЕ УСТАНОВЛЕНЫ (это зависит от инсталляции). В этом случае в списке сервисов плейбука нужно закомментировать те из них, для которых кеш-кластеры Redis не были развернуты.

- Проверить очистку кешей Redis:
  - Зайти на любой **dos**-сервер с Redis и выполнить команду:

```
docker exec -it redis_minos redis-cli -p 6377 --no-auth-warning -a <пароль> info keyspace
```

  - Убедиться, что в выводе команды нет лишних символов, пробелов и т. п. (только слово `keyspace`); созданный ранее ключ должен быть удален.

10.Настроить синхронизацию поиска ЦОД2 из ЦОД1

Сервисы в ЦОД1 работают в своем обычном режиме и не требуют дополнительной настройки для передачи данных в ЦОД2.

- Необходимо изменить конфигурационные файлы и перезапустить контейнеры mailbek и dirbek, для этого выполнить плейбук следующей командой:

```
ansible-playbook -i inventory/ucs_disaster02_msk.yml playbooks/mailion/disaster/disaster_search_replace_sync.yml --extra-vars "ansible_user=astra" -v
```

Этот плейбук меняет в конфигурации сервисов mailbek_search и dirbek строки "service": "search_replic" на "service": "search_sync" (для mailbek_search) и "service": "dir_replic" на "service": "dir_sync" (для dirbek) и перезапускает контейнеры, тем самым включая режим синхронизации.

- Проверить работу в режиме репликации:

```
ssh ucs-search-1.mailion-disaster-02.msk.example.com
ls -la /srv/docker/mailbek_search/data/index/
```

Должны появиться не менее шести файлов с актуальными отметками времени в имени.

- После переключения проверить конфигурацию сервисов **dirbek** и **mailbek_search** на поисковых машинах:

```
ssh ucs-search-1.mailion-disaster-02.msk.example.com
sudo docker ps -a | grep mailbek_search
sudo docker ps -a | grep dirbek
sudo cat /srv/docker/mailbek_search/conf/config.json | grep "service"
```

В выводе должна присутствовать строка: `"service": "search_sync"`.

```
sudo cat /srv/docker/dirbek/conf/config.json | grep "service"
```

В выводе должна присутствовать строка: `"service": "dir_sync"`.

- Также для проверки работоспособности проверить журналы **mailbek_search** и **dirbek**:

```
ssh ucs-infra-1.mailion-disaster-02.msk.example.com
sudo less /srv/logs/ucs_disaster/mailbek_search/mailbek_search.log
sudo less /srv/logs/ucs_disaster/dirbek/dirbek.log
```

Если в журналах присутствует следующее сообщение об ошибке, это нормально:

```
Jul  5 18:53:26 ucs-search-2.mailion-disaster-02.msk.example.com
mailbek_search[1]: 2024-07-05T18:53:26.493+0000 ERROR
omicron/src/replica/actions.cpp:50  GetBucket() error   {"service":"mailbek-
search","version":"1.43.5","service_endpoint":"mailbek-search.ucs-search-
2.mailion-disaster-02.msk.example.com","docid":"---unique-index-stats-
document---","code":"14","info":"document has no quotation image"}
```

В конфигурационных файлах поиска период репликации задается, соответственно для **maiblek** и **dirbek**, в секциях **search_sync** и **dir_sync**, с помощью параметра `search_sync->upgrade→period` и по умолчанию равен "**10m**".

11.Настроить репликацию DOS

Общая информация по логике работы DOS при репликации

- В ЦОД1 DOS переключается в режим обслуживания (maintenance), а в ЦОД2 — в режим ожидания (standby), после этого в ЦОД1 создаются резервные копии.
- В RocksDB в файлах собственного формата хранятся метаданные DOS. Для этих файлов (только на время работы DOS в режиме обслуживания) создаются символические ссылки (symlink).

- Затем DOS автоматически выводится из режима обслуживания, после этого файлы копируются в фоновом режиме, а БД в ЦОД1 доступна и на запись, и на чтение.
- Объем данных большой, копирование занимает много времени. Все файлы доступны только для чтения. Изменения пишутся только в новые файлы, а старые всегда доступны только для чтения.
- Операция резервного копирования требует остановки системы только на время создания резервных копий. Во время резервного копирования хранилище DOS будет доступно только для чтения.

Для настройки репликации необходимо создать резервные копии в ЦОД1 и запустить DOS в ЦОД2 в режиме восстановления из резервных копий. Для этого есть плейбук.

- Подготовить inventory-файл: поменять хосты на свои, вставить открытый ssh-ключ. Пример:

```
#Inventory used by nct.dispersed_object_store.crossdc_init role
---

all:
  children:
    dispersed_object_store:
       children:
          #main cluster
          ucs_dos_shard:
            hosts:
              ucs-obst-1.mailion-disaster-01.msk.example.com:
                dos_node_id: 1
              ucs-obst-2.mailion-disaster-01.msk.example.com:
                dos_node_id: 2
              ucs-obst-3.mailion-disaster-01.msk.example.com:
                dos_node_id: 3
              ucs-obst-4.mailion-disaster-01.msk.example.com:
                dos_node_id: 4
            vars:
              dispersed_object_store_cluster_role: main

          #secondary cluster
          ucs_dos_shard2:
            hosts:
              ucs-obst-1.mailion-disaster-02.msk.example.com:
                dos_node_id: 1
              ucs-obst-2.mailion-disaster-02.msk.example.com:
                dos_node_id: 2
              ucs-obst-3.mailion-disaster-02.msk.example.com:
                dos_node_id: 3
              ucs-obst-4.mailion-disaster-02.msk.example.com:
                dos_node_id: 4
            vars:
              dispersed_object_store_cluster_role: secondary

  vars:
    ansible_ssh_user: "astra"
    vault_secret_path: "mailion/data/stages/dos_shard/installation"
    ssh_private_key_path: "/root/.ssh/awx.key"
    authorized_keys:
      root:
# Тут вставить открытый ssh-ключ
        "awx": >-
          ssh-rsa
          <код ключа>
          root@awx.example.com
```

Рекомендуется применять плейбук, используя такой же закрытый ключ, что хранится в Vault по указанному в параметре `vault_secret_path` пути.

- Выполнить команду:

```
ansible-playbook playbooks/crossdc_repl.yml \
        -e private_ansible_user=astra \
        --diff \
        -b \
        --private-key ~/.ssh/myoffice/awx -i inventory/ucs_cross_dc.yml
```

На текущий момент работа плейбука возможна только при наличии Vault.

> В процессе работы плейбука происходит копирование резервных копий из ЦОД1 в ЦОД2 напрямую — с машины на машину, на тот же диск, где находятся данные DOS в ЦОД2. При большом объеме данных эта операция может занимать много времени, следует это учитывать.

- Итоговое состояние:
  - DOS в ЦОД1 — режим **normal** (доступно для записи).
  - DOS в ЦОД2 — режим **standby** (доступны только запросы к API для чтения), постоянная синхронизация с ЦОД1 в фоновом режиме.
- Проверить репликацию DOS.
  - На любой ноде с DOS в ЦОД1 зайти в docker-контейнер **dispersed_objsect_store** с помощью следующей команды (флаг `-l` в конце команды важен для подгрузки переменных среды):

```
docker exec -it dispersed_object_store bash -l
```

  - Выполнить внутри контейнера команду:

```
ucs-dispersed-object-store-client api write -n mail -k kek1 -d bar
```

  - На любой ноде с DOS в ЦОД2 зайти в docker-контейнер, как написано выше для ЦОД1.
  - Выполнить внутри контейнера команду:

```
ucs-dispersed-object-store-client api read -n mail -k kek1
```

  - В выводе должно присутствовать значение «`bar`» (либо, если было задано другое, то другое значение).

12. **Удостовериться, что в конфигурационных файлах в ЦОД1 и ЦОД2 задано одно и то же значение region_id**.

Если по каким-либо причинам значения **region_id** будут отличаться, то это различие необходимо устранить (см. Предварительная настройка (до репликации)).

13. **Удостовериться, что в конфигурационных файлах в ЦОД1 и ЦОД2 задано одно и то же значение SSID для house фронтенд-сервиса**.

Если по каким-либо причинам значения **SSID** будут отличаться, то это различие необходимо устранить (см. Предварительная настройка (до репликации)).

5.3 Воспроизведение катастрофы

1. Перед проверкой необходимо убедиться, что основной функционал Mailion

работает.

См. Сценарии проверки инсталляций

2. **Выключить виртуальные машины в ЦОД1.**

- Выполнить команду:

```
ansible -i inventory/ucs_disaster01_msk.yml ucs -m ansible.builtin.shell -a "sudo
shutdown -h now
```

- Для проверки, что все ВМ остановились выполнить команду:

```
ansible all -m ping -i inventory/ucs_disaster01_msk.yml -e
private_ansible_user=astra --private-key=~/.ssh/myoffice/awx
```

3. **Убедиться, что MongoShake не пишет в ЦОД2:**

- По умолчанию для параметра **restartPolicy** в MongoShake задается значение **no**, т.е. после перезапуска машины в ЦОД1, сервис не запускается.

4. **Переключить сервисы поиска в ЦОД2 из режима синхронизации с ЦОД1 в режим репликации (кластерный режим работы) с помощью плейбука:**

```
ansible-playbook -i inventory/ucs_disaster02_msk.yml
playbooks/mailion/disaster/disaster_search_replace_replic.yml --extra-vars
"ansible_user=astra" -v \
    --private-key ~/.ssh/myoffice/awx \
    -e private_ansible_user=astra
```

- После переключения проверить конфигурацию сервисов **mailbsek_search** и **dirbek** на поисковых машинах (в выводе должна присутствовать строка `"service": "search_replic"`):

```
ssh ucs-search-1.mailion-disaster-02.example.com
sudo docker ps -a | grep mailbek_search
sudo docker ps -a | grep dirbek
sudo cat /srv/docker/mailbek_search/conf/config.json | grep "service"
```

  - В выводе должна присутствовать строка `"service": "search_replic"`.

```
sudo cat /srv/docker/dirbek/conf/config.json | grep "service"
```

  - В выводе должна присутствовать строка `"service": "dir_replic"`.
  - Также необходимо проверить журналы **mailbek_search** и **dirbek**:

```
ssh ucs-infra-1.mailion-disaster-02.example.com
sudo less /srv/logs/ucs_disaster/mailbek_search/mailbek_search.log
sudo less /srv/logs/ucs_disaster/dirbek/dirbek.log
```

5. **Выключить репликацию DOS на ЦОД2.**

   Если оставить репликацию DOS включенной в ЦОД2 и в ЦОД1, то при включении DOS в ЦОД1, изменения, сделанные в ЦОД2, будут добавлены автоматически. **Но это сработает, только если на время катастрофы (ЦОД2 стал основным, а ЦОД1 восстанавливается) в журнале операций DOS не произошла ротация.** Также возможна ситуация, когда в ЦОД1 будет потеряна значительная часть данных или даже все данные (когда данные окажутся за пределами журнала операций). Поэтому рекомендуется всегда выключать репликацию в ЦОД2 (резервном кластере) при аварийном переключении в случае катастрофы, и при обратном переключении на ЦОД1 (восстановленный основной кластер) восстановить DOS из резервной копии (см. раздел Обратное переключение).

- На любой машине с DOS в ЦОД2:

```
ssh ucs-obst-1.mailion-disaster-02.example.com
```

- Зайти в контейнер DOS:

```
sudo docker exec -it dispersed_object_store bash -l
```

- Выполнить команду отключения репликации:

```
ucs-dispersed-object-store-client dc delete --force
```

- Проверить выключение репликации DOS в ЦОД2 с помощью следующей команды внутри контейнера на любом экземпляре DOS:

```
ucs-dispersed-object-store-client leader get_state
```

В выводе после выполнения команды должна быть секция:

```
"cross_dc": {
            "clusters": {},
            "version": 0,
            "settings": null,
            "topology": null
        }
```

6. **Перевести кластер DOS в ЦОД2 в режим normal (разрешение записи).**

При настройке репликации DOS кластер переходит в режим ожидания (**standby**). В этом режиме DOS не может выполнять запросы на запись, поэтому необходимо вручную перевести его в режим **normal**.

- Зайти на любую ноду DOS:

```
ssh ucs-obst-1.mailion-disaster-02.example.com
```

- Зайти в контейнер:

```
sudo docker exec -it dispersed_object_store bash -l
```

- Выполнить команду:

```
ucs-dispersed-object-store-client leader set_cluster_mode --mode=NORMAL
```

7. Запустить сервисы Mailion в ЦОД2.

- Выполнить команду:

```
ansible-playbook \
    -i inventory/ucs_disaster02_msk.yml
playbooks/mailion/disaster/disaster_start_services.yml  \
    --private-key ~/.ssh/myoffice/awx \
    -e private_ansible_user=astra
```

- Проверить запуск сервисов с помощью Kunkka: убедиться, что сервисы имеют статус READY.

8. Переключить DNS.

- Поменять IP-адрес DNS для следующих записей:

```
disaster-msk IN A <ip_vip_frontend_disaster_msk02>
disaster-msk IN MX 10 mx1-disaster-example.com.
disaster-msk IN MX 20 mx2-disaster-example.com.
mx1-disaster-msk IN A <ip_mail1_disaster_msk02>
mx2-disaster-msk IN A <ip_mail2_disaster_msk02>
```

Пример записей в текущий момент.

ЦОД1:

```
10.20.35.216 ucs-mail-1.mailion-disaster-01.example.com
10.20.34.155 ucs-mail-2.mailion-disaster-01.example.com
10.20.34.139 ucs-frontend-1.mailion-disaster-01.example.com
```

ЦОД2:

```
10.20.36.145 vip ip ucs-frontend-1.mailion-disaster-02.example.com
10.20.33.153  ucs-mail-1.mailion-disaster-02.example.com
10.20.35.71 ucs-mail-2.mailion-disaster-02.example.com
```

- Выполнить проверку. На локальной машине выполнить команду **host**:

```
$ host mx1-disaster-msk.stageoffice.ru
mx-disaster-example.com has address 10.20.33.153
$ host mx2-disaster-example.com
mx-disaster-example.com has address 10.20.35.71
$ host disaster-example.com
disaster-example.com has address 10.20.36.145
```

9. Проверить созданные в ЦОД1 объекты на ЦОД2.

См. Сценарии проверки инсталляций.

10. Создать объекты в ЦОД2.

См. Сценарии проверки инсталляций.

5.4 Настройка обратной репликации

Практически все пункты в этом разделе повторяют аналогичные из раздела Настройка репликации, но выполняются в отношении ЦОД1.

Начальные условия:

- ЦОД1 восстановлен и готов к эксплуатации (виртуальные машины остановлены).
- Кластер ЦОД2 является основным, ЦОД1 — резервным (после катастрофы).
- DNS-сервер настроен на ЦОД2, куда и направлен весь трафик.
- Репликация между кластерами отсутствует.
- Все машины в ЦОД1 остановлены.

1. **Запустить ВМ в ЦОД1 и проверить ее доступность:**

```
ansible all -m ping -i inventory/ucs_disaster01_msk.yml -e
private_ansible_user=astra --private-key=~/.ssh/myoffice/awx
```

2. **Проверить наличие сертификатов на каждой ноде DOS в ЦОД1** (см. этап копирования сертификатов в разделе Настройка репликации)**.**

```
ssh ucs-obst-1.mailion-disaster-01.example.com
ls -la /srv/tls/certs/ucs-infra-1.mailion-disaster-01.example.com-main-ca.pem
```

3. **Проверить наличие сертификатов поиска в ЦОД1** (см. этап копирования сертификатов в разделе Настройка репликации)**.**

4. **Проверить наличие сертификатов на ноде, где запущена утилита MongoShake** (см. этап копирования сертификатов в разделе Настройка репликации).

5. **Создать резервные копии БД MongoDB в каждом ЦОД**

- На машине **infra** каждого ЦОД вручную запустить **bash**-скрипт для создания резервных копий (включен в поставку):

```
/srv/docker/mongodb/backup_scripts/mongodb_backup_generic.sh.
```

- Сохранить резервные копии в любое удобное место.

В поставку также включена **cron**-задача, которая по умолчанию делает резервные копии MongoDB каждый день в 01:00 при условии, что во время установки Mailion она не была выключена и был установлен плейбук для резервного копирования.

6. **Создать резервную копию MongoDB в ЦОД2 для будущего восстановления в ЦОД1 перед запуском MongoShake**. Резервную копию в ЦОД1 можно создать так же, как описано в предыдущем разделе.

7. **Остановить сервисы в ЦОД1**

- Выполнить команду:

```
ansible-playbook \
    -i inventory/ucs_disaster01_msk.yml
playbooks/mailion/disaster/disaster_stop_services.yml  \
    --private-key ~/.ssh/myoffice/awx \
    -e private_ansible_user=astra
```

- С помощью Kunkka проверить, что остались включенными **hydra**, **mailbek**, **dirbek**, **dispersed-object-store**, **mongodb**, сервисы **infra**. Также для проверки следует зайти на ноды, где запущены указанные сервисы и проверить их статус с помощью команды `docker ps`.

8. Очистить БД MongoDB в ЦОД1

Этот шаг необходим, чтобы на этапе запуска MongoShake не было конфликтов при записи данных из ЦОД2 в ЦОД1.

Настроить подключение к мастер-узлу набора реплик (*master replica set*) через стандартный клиент **MongoDB** — **mongosh**, указав логин и пароль, которые использовались при установке и развертывании Mailion:

```
mongosh \
--shell \
--tlsCertificateKeyFile ./certs/merged_mongodb.ucs-db-1.mailion-disaster-
01.msk.stageoffice.ru-main-peer.pem \
--tlsCAFile ./certs/ucs-infra-1.mailion-disaster-01.msk.stageoffice.ru-main-ca.pem
\
--tlsAllowInvalidHostnames \
--tls \
--verbose \
--username <пользователь> \
--password <пароль> \
--port 27017 \
--host mongodb.ucs-db-3.mailion-disaster-01.msk.stageoffice.ru
```

Выполнить команды:

```
show dbs
use admin;
db.getMongo().getDBNames().filter(n => !
['admin','local','config'].includes(n)).forEach(dname =>
db.getMongo().getDB(dname).dropDatabase());
show dbs
```

9. Удалить из MongoDB в ЦОД2 данные MongoShake.

Этот шаг необходим для сброса контрольных точек с прошлых запусков, чтобы при работе MongoShake учитывался весь журнал операций с самого начала. Если это первая установка, и репликация через MongoShake еще ни разу не настраивалась, этот шаг можно пропустить.

- Использовать процедуру настройки подключения к мастер-узлу набора реплик (*master replica set*) MongoDB из предыдущего пункта, но заменить обозначение ЦОД1 на ЦОД2).
- Выполнить команды:

```
use mongoshake
db.dropDatabase('mongoshake')
```

10. **Восстановить MongoDB в ЦОД1 из созданной ранее резервной копии в ЦОД2, используя bash-скрипт.**

    Текст **bash**-скрипта:

```
#!/bin/bash -xe

BACKUP_FILE="mongodb_dump_2024_06_28_1710.gz"
BACKUP_DIR="/srv/backups/mongodb/"
MONGODB_ROOT_PASSWORD="DB_PASS"

DB_LIST=(
  "achill" \
    "beef" \
    "clotho" \
    "dafnis" \
    "daidal" \
    "dorofej" \
    "ektor" \
    "erakles" \
    "eratosthenis" \
    "euripides" \
    "hog" \
    "homeros" \
    "kongur" \
    "kronos" \
    "marker" \
    "odusseus" \
    "perseus" \
    "sophokles" \
    "talaos" \
    "themis" \
    "theseus" \
    "thoth"
)

for db in ${DB_LIST[@]}; do
        echo $db;
        docker run -it --rm -v ${BACKUP_DIR}:/backups -
v /srv/tls/certs:/etc/pki/tls/certs \
        -e 'MONGO_CONN=mongodb://root:${MONGODB_ROOT_PASSWORD}@mongodb.ucs-db-
1.mailion-disaster-01.msk.stageoffice.ru:27017,mongodb.ucs-db-2.mailion-disaster-
01.msk.stageoffice.ru:27017,mongodb.ucs-db-3.mailion-disaster-
01.msk.stageoffice.ru:27017/?
authSource=admin&replicaSet=ucs&tls=true&tlsCAFile=/etc/pki/tls/certs/ucs-infra-
1.mailion-disaster-01.msk.stageoffice.ru-main-
ca.pem&tlsCertificateKeyFile=/etc/pki/tls/certs/merged_mongodb.ucs-db-1.mailion-
disaster-01.msk.stageoffice.ru-main-peer.pem' \
        --name mongorestore hub.stageoffice.ru/mongo:6.0.14-32 \
        sh -c "mongorestore --drop  --gzip \$MONGO_CONN --
authenticationDatabase=admin --nsInclude=\"$db.*\" --maintainInsertionOrder --
numParallelCollections=1 --numInsertionWorkersPerCollection=1 --objcheck --
convertLegacyIndexes --stopOnError  --archive=/backups/${BACKUP_FILE}"
done
```

В скрипте подставить логин и пароль для MongoDB, которые использовались при установке Mailion. Помимо этого, можно поменять имя файла резервной копии и путь к нему, а также пути к файлам с сертификатами.

Необходимо оставить следующие параметры равными 1:
`--numParallelCollections=1`
`--numInsertionWorkersPerCollection=1`

В скрипте восстановление выполняется для каждой указанной базы в отдельности. Это нужно, чтобы избежать ошибок утилиты при восстановлении.

Пропускать этот шаг нельзя, иначе старые данные из MongoDB в ЦОД1 не попадут в MongoDB в ЦОД2 из-за возможной ротации журнала операций.

Запустить скрипт на машине **infra** в ЦОД2.

11.Запустить MongoShake

- Подготовить inventory-файл:

```
all:
  children:

### SECTION 1: grouping by Roles
    ## App: UCS
    ucs:
      children:
        ucs_mongodb_target:
          hosts:
            ucs-db-[1:3].mailion-disaster-01.msk.stageoffice.ru:

        ucs_mongodb_source:
          hosts:
            ucs-db-[1:3].mailion-disaster-02.msk.stageoffice.ru:

        ucs_infrastructure:
          hosts:
            ucs-infra-1.mailion-disaster-02.msk.stageoffice.ru:

## SECTION 2: grouping by tier
    ucs_disaster_02_msk:
      # all servers should be listed
      hosts:
        ucs-infra-1.mailion-disaster-02.msk.stageoffice.ru:
        ucs-db-[1:3].mailion-disaster-01.msk.stageoffice.ru:
        ucs-db-[1:3].mailion-disaster-02.msk.stageoffice.ru:
      vars:
        tier: "ucs_disaster_02_msk"
```

- Задать в **group_vars** следующие переменные:

```
# Mongoshake
tls_certs_dir: "/srv/tls/certs"
tls_keys_dir: "/srv/tls/keys"

mongoshake_source_db_user: "root"
mongoshake_source_db_password: "SOURCE_MONGODB_ROOT_PASSWORD"
mongoshake_source_db_hosts: "{{ groups['ucs_mongodb_source'] }}"

mongoshake_target_db_user: "root"
mongoshake_target_db_password: "TARGET_MONGODB_ROOT_PASSWORD"
mongoshake_target_db_hosts: "{{ groups['ucs_mongodb_target'] }}"

mongoshake_target_db_tls:
  tls_enabled: true
  root_ca_file: "ucs-infra-1.mailion-disaster-01.msk.stageoffice.ru-main-ca.pem"
  merged_pem_file: "merged_mongodb.ucs-db-1.mailion-disaster-
01.msk.stageoffice.ru-main-peer.pem"
  client_cert: "sophokles.ucs-apps-1.mailion-disaster-01.msk.stageoffice.ru-main-
client.pem"
  client_key: "sophokles.ucs-apps-1.mailion-disaster-01.msk.stageoffice.ru-main-
key.pem"
mongoshake_source_db_tls:
  tls_enabled: true
  root_ca_file: "ucs-infra-1.mailion-disaster-02.msk.stageoffice.ru-main-ca.pem"
  merged_pem_file: "merged_mongodb.ucs-db-1.mailion-disaster-
02.msk.stageoffice.ru-main-peer.pem"
  client_cert: "sophokles.ucs-apps-1.mailion-disaster-02.msk.stageoffice.ru-main-
client.pem"
  client_key: "sophokles.ucs-apps-1.mailion-disaster-02.msk.stageoffice.ru-main-
key.pem"

mongoshake_container_state: "stopped"
mongoshake_restart_policy: "no"

mongoshake_command:
  - "-conf"
  - "/apps/collector.conf"
  - "-verbose"
  - "2"
```

- Выполнить команду

```
ansible-playbook playbooks/mailion/disaster/mongoshake.yml \
        -e private_ansible_user=astra \
        --diff \
        --private-key ~/.ssh/myoffice/ldap -i
inventory/ucs_mongoshake_disaster_msk.yml
```

- Запустить контейнер MongoShake
  - Зайти на машину, на которой создан контейнер MongoShake:

```
ssh ucs-infra-1.mailion-disaster-02.msk.stageoffice.ru
```

  - Запустить скрипт:

```
/srv/docker/mongoshake/conf/start_mongoshake_vault.sh
```

  - Ввести пароль от исходной БД MongoDB в ЦОД2 и от целевой БД MongoDB в ЦОД1.
- Проверить ход репликации с помощью MongoShake:

```
http://ucs-infra-1.mailion-disaster-01.stageoffice.ru:49100/repl
```

12.Выполнить сброс Redis в ЦОД1

После завершения установки и проверок в Redis остаются некоторые закешированные данные, которые могут помешать пользователям авторизоваться, когда ЦОД1 станет активным (сервис **minos**), и также могут мешать запуску сервисов (в журналах появляется сообщение об ошибке «*minos: right ids must not be empty*»).

- Запустить плейбук:

```
ansible-playbook -i inventory/ucs_disaster01_msk.yml
playbooks/mailion/disaster/redis_clear_cache.yml --extra-vars "ansible_user=astra"
-e redis_clear_cached_enabled=true -v
```

Для некоторых сервисов при установке Mailion кеш-кластеры Redis могут быть НЕ установлены (это зависит от инсталляции). В этом случае в списке сервисов плейбука нужно закомментировать те из них, для которых кеш-кластеры Redis не были развернуты.

- Проверить очистку кешей Redis:
  - Зайти на любой **dos**-сервер с **Redis** и выполнить команду:

```
docker exec -it redis_minos redis-cli -p 6377 --no-auth-warning -a <пароль> info
keyspace
```

  - Убедиться, что в выводе команды нет лишних символов, пробелов и т. п. (только слово `keyspace`); созданный ранее ключ должен быть удален.

13.Настроить синхронизацию поиска ЦОД1 из ЦОД2

- Необходимо изменить конфигурационные файлы и перезапустить контейнеры **mailbek** и **dirbek**, для этого выполнить плейбук следующей командой:

```
ansible-playbook -i inventory/ucs_disaster01_msk.yml
playbooks/mailion/disaster/disaster_search_replace_sync.yml --extra-vars
"ansible_user=astra" -v
```

Этот плейбук меняет в конфигурации сервисов **mailbek_search** и **dirbek** строки `"service": "search_replic"` на `"service": "search_sync"` (для **mailbek_search**) и `"service": "dir_replic"` на `"service": "dir_sync"` (для **dirbek**) и перезапускает контейнеры, тем самым включая режим синхронизации.

- Проверить работу в режиме репликации:

```
ssh ucs-search-1.mailion-disaster-02.msk.stageoffice.ru
ls -la /srv/docker/mailbek_search/data/index/
```

Должны появиться не менее шести файлов с актуальными отметками времени в имени.

- Проверить конфигурацию сервисов **dirbek** и **mailbsek_search** на поисковых машинах:

```
ssh ucs-search-1.mailion-disaster-01.msk.stageoffice.ru
sudo docker ps -a | grep mailbek_search
sudo docker ps -a | grep dirbek
sudo cat /srv/docker/mailbek_search/conf/config.json | grep "service"
```

В выводе должна присутствовать строка: `"service": "search_sync"`.

```
sudo cat /srv/docker/dirbek/conf/config.json | grep "service"
```

В выводе должна присутствовать строка: `"service": "dir_sync"`.

- Проверить журналы **mailbek_search** и **dirbek**:

```
ssh ucs-infra-1.mailion-disaster-02.msk.stageoffice.ru
sudo less /srv/logs/ucs_disaster/mailbek_search/mailbek_search.log
sudo less /srv/logs/ucs_disaster/dirbek/dirbek.log
```

14.Настроить репликацию DOS

- Подготовить inventory-файл: поменять хосты на свои, вставить открытый ssh-ключ. Пример:

```
#Inventory used by nct.dispersed_object_store.crossdc_init role
---

all:
  children:
    dispersed_object_store:
       children:
          #main cluster
          ucs_dos_shard:
            hosts:
              ucs-obst-1.mailion-disaster-01.msk.stageoffice.ru:
                dos_node_id: 1
              ucs-obst-2.mailion-disaster-01.msk.stageoffice.ru:
                dos_node_id: 2
              ucs-obst-3.mailion-disaster-01.msk.stageoffice.ru:
                dos_node_id: 3
              ucs-obst-4.mailion-disaster-01.msk.stageoffice.ru:
                dos_node_id: 4
            vars:
              dispersed_object_store_cluster_role: secondary

          #secondary cluster
          ucs_dos_shard2:
            hosts:
              ucs-obst-1.mailion-disaster-02.msk.stageoffice.ru:
                dos_node_id: 1
              ucs-obst-2.mailion-disaster-02.msk.stageoffice.ru:
                dos_node_id: 2
              ucs-obst-3.mailion-disaster-02.msk.stageoffice.ru:
                dos_node_id: 3
              ucs-obst-4.mailion-disaster-02.msk.stageoffice.ru:
                dos_node_id: 4
            vars:
              dispersed_object_store_cluster_role: main

  vars:
    ansible_ssh_user: "astra"
    vault_secret_path: "mailion/data/stages/dos_shard/installation"
    ssh_private_key_path: "/root/.ssh/awx.key"
    authorized_keys:
      root:
# Тут вставить свой открытый ssh-ключ
        "awx": >-
          ssh-rsa
          <код ключа>
          root@awx.stageoffice.ru
```

Рекомендуется применять плейбук, используя такой же закрытый ключ, что хранится в Vault по указанному в параметре `vault_secret_path` пути.

- Выполнить команду:

```
ansible-playbook playbooks/crossdc_repl.yml \
        -e private_ansible_user=astra \
        --diff \        -b \
        --private-key ~/.ssh/myoffice/awx -i inventory/ucs_cross_dc.yml
```

На текущий момент работа плейбука возможна только при наличии Vault.

В процессе работы плейбука происходит копирование резервных копий из ЦОД1 в ЦОД2 напрямую — с машины на машину, на тот же диск, где находятся данные

> DOS в ЦОД2. При большом объеме данных эта операция может занимать много времени, следует это учитывать.

- Итоговое состояние:
  - Хранилище DOS в ЦОД2 — режим **normal** (доступно для записи).
  - Хранилище DOS в ЦОД1 — режим **standby** (общедоступный API, доступно только для чтения), постоянная синхронизация с ЦОД1 в фоновом режиме.

- Проверить репликацию DOS.
  - На любой ноде с DOS в ЦОД2 зайти в docker-контейнер **dispersed_objsect_store** с помощью следующей команды (флаг `-l` в конце команды необходим для подгрузки переменных среды):

```
docker exec -it dispersed_object_store bash -l
```

  - Выполнить внутри контейнера команду:

```
ucs-dispersed-object-store-client api write -n mail -k kek1 -d bar
```

  - На любой ноде с DOS в ЦОД1 зайти в docker-контейнер, как написано выше для ЦОД2.
  - Выполнить внутри контейнера команду:

```
ucs-dispersed-object-store-client api read -n mail -k kek1
```

  - В выводе должно присутствовать значение «`bar`» (либо, если было задано другое, то другое значение).

14. **Удостовериться, что в обоих конфигурационных файлах в ЦОД1 и ЦОД2 задано одно и то же значение region_id.**

Если по каким-либо причинам значения **region_id** будут отличаться, то это различие необходимо будет устранить (см. Предварительная настройка (до репликации)).

15. **Удостовериться, что в обоих конфигурационных файлах в ЦОД1 и ЦОД2 задано одно и то же значение SSID для house фронтенд-сервиса.**

Если по каким-либо причинам значения SSID будут отличаться, то это различие необходимо будет устранить (см. Предварительная настройка (до репликации)).

5.5 Обратное переключение

Начальные условия:

- ЦОД1 — резервный, ЦОД2 — основной.

- Настроена репликация из ЦОД2 в ЦОД1.

1. Остановить сервисы в ЦОД2 (резервном).

- Выполнить скрипт на машине оператора:

```
ansible-playbook \
    -i inventory/ucs_disaster02_msk.yml
playbooks/mailion/disaster/disaster_stop_services.yml  \
    --private-key ~/.ssh/myoffice/awx \
    -e private_ansible_user=astra
```

- Проверить остановку сервисов с помощью Kunkka.

2. Переключить поиск.

- Выполнить команду:

```
ansible-playbook -i inventory/ucs_disaster01_msk.yml
playbooks/mailion/disaster/disaster_search_replace_replic.yml --extra-vars
"ansible_user=astra" -v
```

- Проверить работу в режиме репликации:

```
ssh ucs-search-1.mailion-disaster-02.example.com
ls -la /srv/docker/mailbek_search/data/index/
```

Должны появиться не менее 6 файлов с актуальными отметками времени в имени.

- Проверить конфигурацию сервисов **dirbek** и **mailbsek_search** на поисковых машинах:

```
ssh ucs-search-1.mailion-disaster-01.example.com
sudo docker ps -a | grep mailbek_search
sudo docker ps -a | grep dirbek
sudo cat /srv/docker/mailbek_search/conf/config.json | grep "service"
```

В выводе должна присутствовать строка: `"service": "search_sync"`.

```
sudo cat /srv/docker/dirbek/conf/config.json | grep "service"
```

В выводе должна присутствовать строка: `"service": "dir_sync"`.

- Проверить журналы **mailbek_search** и **dirbek**:

```
ssh ucs-infra-1.mailion-disaster-01.example.com
sudo less /srv/logs/ucs_disaster/mailbek_search/mailbek_search.log
sudo less /srv/logs/ucs_disaster/dirbek/dirbek.log
```

3. Выключить репликацию DOS в ЦОД1.

- Зайти на любую машину DOS в ЦОД1:

```
ssh ucs-obst-1.mailion-disaster-01.example.com
```

- Зайти в контейнер DOS:

```
docker exec -it dispersed_object_store bash -l
```

- Выполнить команду:

```
ucs-dispersed-object-store-client dc delete --force
```

- Проверить выключение репликации DOS в ЦОД1 с помощью следующей команды внутри контейнера на любом экземпляре DOS:

```
ucs-dispersed-object-store-client leader get_state
```

В выводе должна присутствовать секция:

```
"cross_dc": {
            "clusters": {},
            "version": 0,
            "settings": null,
            "topology": null
        }
```

4. Перевести кластер DOS в ЦОД1 в режим normal (разрешение записи).

При настройке репликации кластер DOS переходит в режим **standby**. В этом режиме DOS не может выполнять запросы на запись, поэтому необходимо вручную перевести его в режим **normal**.

- Зайти на любую ноду DOS:

```
ssh ucs-obst-1.mailion-disaster-01.example.com
```

- Зайти в контейнер:

```
sudo docker exec -it dispersed_object_store bash -l
```

- Выполнить команду:

```
ucs-dispersed-object-store-client leader set_cluster_mode --mode=NORMAL
```

5. Выключить MongoShake.

- Зайти на ноду с MongoShake:

```
ssh loader.mailion-disaster-01.example.com
```

- Выполнить команду:

```
sudo docker stop mongoshake && sudo docker rm mongoshake
```

6. Запустить сервисы в ЦОД1.

- Выполнить команду:

```
ansible-playbook \
    -i inventory/ucs_disaster01_msk.yml
playbooks/mailion/disaster/disaster_start_services.yml  \
    --private-key ~/.ssh/myoffice/awx \
    -e private_ansible_user=astra
```

- Проверить в Kunkka, что сервисы имеют статус READY.

7. Переключить DNS (см. таблицу записей DNS выше).

- Для записи `disaster-msk IN A` изменить адрес с 10.20.36.145 на 10.20.34.139 (запись `disaster-msk` должна ссылаться на фронтенд машины ЦОД1).

- Для одной из записей `mx1-disaster-msk IN MX` изменить адрес с 10.20.35.71 на 10.20.35.216 (запись должна ссылаться на почтовый сервер ЦОД1).
- Адрес второй записи `mx2-disaster-msk IN MX` изменить на 10.20.34.155 (запись должна ссылаться на другой почтовый сервер ЦОД1).

Определить IP-адрес хоста можно с помощью команд **host**, **dig**, **ping** и т.д. как описано ранее в документации.

8. **Проверить работоспособность ЦОД1.**

   См. Сценарии проверки инсталляций.

5.6 Сценарии проверки инсталляций

Настройка катастрофоустойчивости

Проверяемые компоненты:

- MongoDB (создать пользователя, создать календарь, событие);
- поиск (отправить письмо, наличие подсказки при вводе в поле «Кому»);
- DOS (отправить письмо с вложением).

Сценарий проверки:

1. Создать пользователя **user1**.
2. Создать администратора тенанта.
3. Под логином пользователя **user1** отправить администратору тенанта письмо с темой **subject1**, текстом **body1** и вложением (картинкой).
4. Под логином администратора тенанта:
   - проверить входящую почту (открыть письмо **subject1**-**body1**, открыть вложение);
   - найти поиском письмо по тексту **body1** во входящих.
5. Под логином пользователя **user1** создать событие **event1**.
6. Проверить по журналам поиска выполнение репликации объектов.

Этап: до катастрофы (переключения)

Проверяемые компоненты:

- MongoDB (создать пользователя, создать календарь, событие);
- поиск (отправить письмо, наличие подсказки при вводе в поле «Кому»);

- DOS (отправить письмо с вложением).

Сценарий проверки создания объектов в ЦОД1:

1. Создать пользователя **user1**.
2. Создать администратора тенанта.
3. Под логином пользователя **user1** отправить администратору тенанта письмо с темой **subject1**, текстом **body1** и вложением (картинкой).
4. Под логином администратора тенанта:
   - проверить входящую почту (открыть письмо **subject1-body1**, открыть вложение);
   - найти поиском письмо по тексту **body1** во входящих.
5. Под логином пользователя **user1** создать событие **event1**.
6. Проверить по журналам поиска выполнение репликации объектов.

Этап: после катастрофы (переключения)

Сценарий проверки перемещения объектов в ЦОД2:

1. Авторизоваться под логином администратора тенанта:
   - проверить наличие письма **subject1-body1** от пользователя **user1** во входящих;
   - проверить наличие и открытие вложения;
   - найти поиском письмо по тексту **body1** во входящих.
2. Авторизоваться под логином пользователя **user1** и проверить наличие события **event1** в календаре.

Сценарий проверки функциональности в ЦОД2:

1. Создать пользователя **user2**.
2. Под логином пользователя **user2**:
   - отправить письмо с темой **subject2**, текстом **body2** и вложением (картинкой) администратору тенанта;
   - создать событие **event2**.
3. Под логином администратора тенанта:
   - проверить поиск пользователя **user2** через административный интерфейс;
   - проверить наличие письма **subject2-body2** от пользователя **user2** во входящих;
   - проверить наличие и открытие вложения письма;
   - найти поиском письмо по тексту **body2** во входящих.

Этап: после восстановления (обратного переключения)

Сценарий проверки перемещения в ЦОД1 объектов, созданных в ЦОД2:

1. Авторизоваться под логином пользователя **user2**.
   - проверить наличие события **event2** в календаре;
2. Авторизоваться под логином администратора тенанта.
   - проверить наличие письма **subject2-body2** от пользователя **user2** во входящих;
   - проверить наличие и открытие вложения письма;
   - найти поиском письмо по тексту **body2** во входящих.

Сценарий проверки создания объектов в ЦОД1:

1. Создать пользователя **user3**.
2. Под логином пользователя **user3**:
   - отправить администратору тенанта письмо с темой **subject3**, текстом **body3** и вложением (картинкой);
   - создать событие **event3**.
3. Под логином администратора тенанта:
   - проверить поиск пользователя **user3** через административный интерфейс;
   - проверить входящую почту (открыть письмо **subject3-body3**, открыть вложение);
   - найти поиском письмо по тексту **body3** во входящих.

6 ТЕХНИЧЕСКАЯ ПОДДЕРЖКА

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

- Адрес электронной почты: support@service.myoffice.ru
- Телефон: 8-800-222-1-888.

7 ИСТОРИЯ ИЗМЕНЕНИЙ ДОКУМЕНТА

Версия 1, дата публикации: 12.11.2024

1. Добавлен раздел «Катастрофоустойчивость».
2. В раздел «Дополнительные возможности и рекомендации по установке» добавлен подраздел «Настройка аудита событий в формате CEF».
3. Добавлен раздел «История изменений документа» (этот раздел).

8 ПРИЛОЖЕНИЕ А. ПРИМЕР НАПИСАНИЯ ВНЕШНИХ DNS-ЗАПИСЕЙ

Имя записи	Пример написания записи
api	api-test.example.com IN A <ucs_frontend_vip>
auth	auth-test.example.com IN A <ucs_frontend_vip>
autoconfig	autoconfig-test.example.com IN A <ucs_frontend_vip>
avatars	avatars-test.example.com IN A <ucs_frontend_vip>
caldav	caldav-test.example.com. 900 IN CNAME <ucs_frontend_vip>
carddav	carddav-test.example.com. 878 IN CNAME <ucs_frontend_vip>
db	db-test.example.com IN A <ucs_frontend_vip>
grpc	grpc-test.example.com IN A <ucs_frontend_vip>
imap	imap-test.example.com IN A <ucs_frontend_vip>
mail	mail-test.example.com IN A <ucs_frontend_vip>
mail._domainkey	mail._domainkey.test.example.com. 899 IN TXT "v=DKIM1;" "g=*;" "k=rsa;" "p=<DKIM_KEY>"
mx1	mx-test.example.com. 900 IN A ucs-mail-1.test.example.com
mx2	mx-test.example.com. 900 IN A ucs-mail-2.test.example.com
preview	preview-test.example.com. 900 IN CNAME <ucs_frontend_vip>
relay	relay-test.example.com. 900 IN A <ucs_mail_vip>
resources	resources-test.example.com. 900 IN A <ucs_frontend_vip>
secured	secured-test.example.com. 900 IN A <ucs_frontend_vip>
smtp	smtp-test.example.com IN A <ucs_mail_vip>
_adsp._domainkey	_adsp._domainkey.test.example.com. 900 IN TXT "dkim=all"

Имя записи	Пример написания записи
_autodiscover._tcp	_autodiscover._tcp.test.example.com. 900 IN SRV 0 0 443 <mailion_external_domain>
_caldavs._tcp	_caldavs._tcp.test.example.com. 900 IN SRV 0 1 6787 caldav-test.example.com.
_carddavs._tcp	_carddavs._tcp.test.example.com. 900 IN SRV 0 1 6787 carddav-test.example.com.
_grpcsec._tcp	_grpcsec._tcp.test.example.com. 900 IN SRV 0 0 3142 grpc-test.example.com.
_imap._tcp	_imap._tcp.test.example.com. 900 IN SRV 10 0 143 imap-test.example.com.
_imaps._tcp	_imaps._tcp.test.example.com. 900 IN SRV 0 0 993 imap-test.example.com.
_smtps._tcp	_smtps._tcp.test.example.com. 900 IN SRV 0 0 465 smtp-test.example.com.
_submission._tcp	_submission._tcp.test.example.com. 900 IN SRV 0 0 587 smtp-test.example.com.
_submissions._tcp	_submissions._tcp.test.example.com. 900 IN SRV 0 0 465 smtp-test.example.com.