МойОфис Профессиональный 3

Руководство по установке

РАБОЧЕЕ ПРОСТРАНСТВО ДЛЯ КОРПОРАТИВНЫХ КОММУНИКАЦИЙ И СОВМЕСТНОЙ РАБОТЫ

СЕРВЕРНАЯ ЧАСТЬ

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

«МОЙОФИС ПРОФЕССИОНАЛЬНЫЙ 3»

РАБОЧЕЕ ПРОСТРАНСТВО ДЛЯ КОРПОРАТИВНЫХ КОММУНИКАЦИЙ И СОВМЕСТНОЙ РАБОТЫ

СЕРВЕРНАЯ ЧАСТЬ

1.5

РУКОВОДСТВО ПО УСТАНОВКЕ

Версия 1

На 72 листах

Дата публикации: 19.06.2024

Москва
2024

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

Товарные знаки «МойОфис» и «MyOffice» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

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

СОДЕРЖАНИЕ

1 Общие сведения ... 8
1.1 Назначение ... 8
1.2 Требования к квалификации персонала ... 8
1.3 Системные требования ... 9
1.3.1 Аппаратные требования ... 9
1.3.1.1 Минимальные требования ... 10
1.3.1.2 Рекомендованные требования ... 10
1.3.2 Программные требования ... 13
1.4 Рекомендации по разметке дисков ... 14
1.5 Требования по работе с DNS ... 14
1.5.1 Организация работы сервисов разрешения имен ... 14
1.5.2 Формирование внешних доменных имен ... 15
1.5.3 Внешние DNS-записи ... 15
1.5.4 Изменение внешних DNS-записей ... 16
1.5.5 Внутренние DNS-записи ... 17
1.6 Ограничения ... 18
1.6.1 Ограничения по работе с файлом inventory ... 18
1.6.2 Ограничения по работе с подсистемой управления конфигурациями ... 18
1.6.3 Ограничения по работе с системами виртуализации ... 18
1.6.4 Ограничения по использованию доменного имени ... 18
1.7 Описание архитектуры продукта ... 19
1.8 Запуск веб-интерфейса ПО ... 19
2 Первичная установка ... 20
2.1 Состав дистрибутива ... 20
2.2 Подготовка к установке ... 20
2.2.1 Аутентификация в docker registry ... 20
2.2.2 Возможные проблемы в работе docker registry ... 21
2.2.3 Описание общих ролей подсистемы Ansible ... 21
2.2.4 Роли, используемые для установки ПО ... 22

2.2.5 Подготовка инфраструктуры установки ... 25
2.2.6 Установка хранилища образов Docker ... 25
2.2.7 Установка подсистемы управления конфигурациями (Ansible) ... 26
2.2.8 Установка ПО сервера оператора ... 27
2.2.9 Размещение SSL-сертификатов для шифрования ... 28
2.2.10 Обновление SSL-сертификатов ... 29
2.2.11 Настройка основных параметров установки ... 29
2.2.11.1 Настройка минимальных параметров установки ... 29
2.2.11.2 Установка системы для работы более 1000 пользователей ... 32
2.2.12 Настройка дополнительных параметров установки ... 32
2.2.13 Настройка межсетевого экранирования ... 33
2.2.14 Настройка службы синхронизации времени NTP ... 34
2.3 Запуск установки ... 34
2.4 Проверка корректности установки ... 34
2.5 Запуск нескольких экземпляров ПО ... 35
2.6 Настройка push-уведомлений в режиме удаленного вызова процедур GRPC ... 35
2.6.1 Ручная настройка ... 36
2.6.2 Автоматическая настройка ... 38
2.6.3 Проверка работы сервиса ... 39
2.6.4 Устранение неполадок ... 39
2.7 Резервное копирование данных ... 39
2.7.1 Резервное копирование MongoDB ... 39
2.7.1.1 Резервное копирование без дополнительного ПО ... 39
2.7.1.2 Использование ПО Percona Backup for MongoDB ... 40
2.7.1.3 Восстановление ... 42
2.7.1.4 Возможные проблемы ... 43
2.7.2 Резервное копирование MinIO ... 43
2.7.3 Применение внесенных изменений в резервное копирование ... 44
2.8 Установка в составе других продуктов «МойОфис» ... 44
3 Обновление с предыдущих версий ... 45
3.1 Обновление системы управления базами данных MongoDB ... 45

3.2 Обновление сервиса поиска Squash до версии 1.10 45

4 Дополнительные возможности и рекомендации по установке 46

4.1 Настройка стенда ПО 46

4.1.1 Добавление стенда на стороне ADFS 47

4.1.2 Известные ограничения 49

4.2 Настройка вебинаров 50

4.2.1 Включение вебинаров 50

4.2.2 Проверка работоспособности вебинаров 51

4.3 Интеграция на примере автоматической телефонной станции Asterisk 52

4.3.1 Процедура настройки взаимодействия компонентов с Asterisk 52

4.3.2 Проверка работоспособности 57

4.4 Настройка виртуальной доски для совместного использования 58

4.4.1 Включение виртуальной доски 58

4.5 Настройка транскрибации речи (субтитры) 59

4.5.1 Системные требования 59

4.5.2 Включение транскрибации речи 60

4.6 Интеграция с Jira 62

4.6.1 Настройка Jira для интеграции с ПО 62

4.6.2 Настройка сервиса squadus_jiratrigger 62

4.6.3 Настройки сервиса jiratrigger 63

4.6.4 Возможные проблемы 64

4.7 Настройка автоматических уведомлений в проекте Jira 64

4.7.1 Настройка входящих webhook 64

4.7.2 Настройка webhook Jira 67

4.8 Режим федерации 68

4.8.1 Настройка режима федерации 69

4.9 Настройка отправки приложений к логам 70

4.10 Добавление стороннего корневого сертификата 71

4.11 Настройка системы для работы более 1000 пользователей 71

4.12 Централизованная установка настольных приложений 72

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

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

Таблица 1 — Сокращения и расшифровки

Сокращение	Расшифровка
ADFS	Active Directory Federation Services — компонент Windows Server обеспечивающий функции провайдера аутентификации для веб-приложений
API	Application Programming Interface — программный интерфейс приложения
Application Service	Служба приложений
АТС	Автоматическая телефонная станция
CPU	Central Processing Unit, процессор
FQDN	Fully Qualified Domain Name — «полностью определенное имя домена» — имя домена, не имеющее неоднозначности в определении. Включает в себя имена всех родительских доменов иерархии DNS
PBM	Percona Backup for MongoDB — распределенное решение с открытым исходным кодом для последовательного резервного копирования и восстановления сегментированных кластеров MongoDB и наборов реплик
SAN	Subject Alternative Name — расширение X.509 позволяющее использовать один сертификат для множества доменов
Webhook	Метод расширения или изменения поведения веб-страницы или веб-приложения с помощью обратных вызовов (в веб-разработке)
ВМ	Виртуальная машина
ООО	Общество с ограниченной ответственностью
ОС	Операционная система
ПО	Программное обеспечение
УЦ	Удостоверяющий центр

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение

Рабочее пространство для корпоративных коммуникаций и совместной работы — безопасный корпоративный мессенджер с поддержкой видеоконференций, глубокой интеграцией с продуктами МойОфис и возможностью интеграции с внешними информационными системами.

В состав продукта входят:

- Коммуникационная система — для обмена мгновенными сообщениями, документами и медиафайлами между пользователями и в групповых чатах;
- Система видеоконференцсвязи (ВКС) — для организации аудио- и видеозвонков и конференций с возможностью гостевого доступа незарегистрированными пользователями;
- Приложения — для рабочего общения с помощью текстовых, голосовых и видео сообщений, а также участия в конференциях в веб-браузерах и на ОС Windows, Linux, macOS, iOS, Android.

Подробное описание возможностей продукта приведено в документе «"МойОфис Профессиональный 3". Функциональные возможности».

1.2 Требования к квалификации персонала

Администратор ПО 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);
   - работа с одной из подсистем серверной виртуализации на базе гипервизоров Hyper-V, VMWare vSphere ESXi, KVM.
3. Опыт работы с командной строкой ОС Linux: знания в объеме курсов Red Hat RH124, RH134, RH254.

4. Опыт работы со службой доменных имен (DNS):
   – знание основных терминов (DNS, IP-адрес и т.д.);
   – понимание принципов работы DNS (корневые серверы, TLD-серверы, серверы имен доменов, разрешающий сервер имен и т.д.);
   – знание типов записи и запросов DNS.
5. 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).
6. Практический опыт администрирования на уровне эксперта:
   – Redis;
   – Prometheus;
   – MongoDB;
   – MinIO.
7. Опыт работы с подсистемой централизованного управления Ansible.
8. Опыт работы со стандартными офисными приложениями.

1.3 Системные требования

1.3.1 Аппаратные требования

В соответствии с архитектурой продукта для оценки ресурсов, выделяемых для поддержания работоспособности, необходимо:

- оценить общее количество пользователей;
- оценить количество одновременно работающих пользователей.

Исходя из полученных результатов, ознакомиться с требованиями к продукту, представленными в разделах «Минимальные требования» и «Рекомендованные требования».

1.3.1.1 Минимальные требования

Минимальные требования для установки серверного и клиентского ПО без режима отказоустойчивости, без системы видеоконференций, приведены в таблицах 2, 3. Требования рассчитаны для общего количества пользователей до 1000, из которых 300 используют ПО одновременно.

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

Таблица 2 — Минимальные требования для установки ПО (Серверное приложение)

Параметр	Значение
Процессор, CPU	8
Оперативная память, Гбайт	16
Дисковая подсистема, Гбайт, тип	100, SSD + возможность увеличения в зависимости от объема данных
Сетевой интерфейс, Мбит/сек	100
Количество серверов	1

Таблица 3 — Минимальные требования для установки ПО (Клиентское приложение)

Параметр	Значение
Процессор, CPU	4
Оперативная память, Гбайт	4
Дисковая подсистема, Мбайт, тип	500, HDD
Операционная система	Windows, Ubuntu LTS, Fedora 28, Debian 8, RHEL 7, MacOS

1.3.1.2 Рекомендованные требования

Рекомендованные требования для установки серверного и клиентского ПО с режимом отказоустойчивости и системой видеоконференций приведены в таблицах 4 и 5. Для всех серверов рекомендовано обеспечить сетевое подключение с пропускной способностью не менее 1 Гбит/сек.

Для достижения лучшей отказоустойчивости не следует совмещать серверы с ролью squadus_st, squadus_db, squaudus_search, squadus_apps

Таблица 4 — Рекомендованные требования для установки ПО (Серверное приложение)

Имя роли сервера	VCPU[1]	RAM, Гбайт	HDD[2], Гбайт (без учета ОС)	SSD[2], Гбайт (без учета ОС)	Количество виртуальных машин	VCPU[1]	RAM, Гбайт	HDD[2], Гбайт	SSD[2], Гбайт
на каждую роль						**итого на группу**			
squadus_ha	2	2	10	0	2	4	4	20	0
squadus_apps	4	8	40	0	2	8	16	80	0
squadus_db	4	16	0	300	3	12	48	0	900
squadus_st	4	8	500	0	3	12	24	1500	0
squadus_search	4	16	300	0	3	12	48	900	0
squadus_mail	2	2	10	0	2	4	4	20	0
squadus_meet_apps	2	4	10	0	2	4	8	20	0
squadus_meet_jibri	4	8	100	0	2	8	16	200	0
squadus_meet_jvb[3]	2 (8)[3]	8	10	0	2	4 (16)[3]	16	20	0
squadus_converter	4	4	10	0	2	8	8	20	0
squadus_redis	2	2	0	5	3	6	6	0	15
squadus_infra	4	8	200	0	1	4	8	200	0
squadus_mq	4	8	40	0	3	12	24	120	0
squadus_meet_vosk	4	16	60	0	1	4	16	60	0
				ИТОГО:	27	86	206	2980	915

[1] Без использования ядер с включенным hyper-threading.

[2] При использовании HDD расчетное количество операций 150-180 IOPS, при использовании SSD — от 200К IOPS.

[3] Для работы в режиме вебинаров конфигурация каждого хоста группы ролей squadus_meet_apps должна быть не менее 8 vCPU и 8Гбайт RAM. Масштабирование количества хостов в группе ролей squadus_meet_jvb при использовании вебинаров должно учитывать соотношение: один хост на 200 участников вебинара.

Таблица 5 — Рекомендованные требования для установки ПО Squadus (Клиентское приложение)

Параметр	**Значение**
Процессор, CPU	4
Оперативная память, Гбайт	4
Дисковая подсистема, Гбайт, тип	500 Гбайт, HDD
Операционная система	MacOS, Windows, Linux, iOS, Android

Описания ролей для серверов, приведеных в таблице 6.

Таблица 6 — Описание ролей серверов

Наименование роли	Описание сервера
squadus_ha	Балансировка нагрузки, между серверами группы используется VRRP адрес
squadus_apps	Приложения, на которых размещены сервисы ПО
squadus_db	Базы данных, используемые для кластера MongoDB
squadus_st	Система хранения данных MinIO
squadus_search	Система поиска и индексации данных
squadus_mail	Электронная почта для рассылки уведомлений от ПО
squadus_meet_apps	Видеоконференция jitsi backend
squadus_meet_jvb	Запись видеоконференций jibri
squadus_meet_jibri	Видеоконференция video bridge
squadus_converter	Обработка прилагаемых файлов в ПО для формирования preview
squadus_redis	Хранение баз данных NoSQL Redis
squadus_infra	Мониторинг и хранения log-файлов сервисов
squadus_meet_vosk	Менеджер очередей NATS
squadus_mq	Сервис субтитров

1.3.2 Программные требования

Требования к ПО для места оператора, на котором производится установка, приведены в таблице 7.

Таблица 7 — Требования к ПО для серверов, на которые производится установка

Требование	Описания	
Поддерживаемые браузеры	Перечень поддерживаемых браузеров приведен в документе «"МойОфис Профессиональный 3". Системные требования»	
Python	Не ниже версии 3.6	
mongosh	Версия 1.8.1	
Модули Python («*» указана минорная версия)	jmespath	0.10.*
	jinja2 [1]	Не ниже версии 3.1.2
	ansible-core	2.11.* (но не выше 2.12)
	netaddr	0.10.*
	dnspython	2.2.*
	passlib	1.7.*
	pymongo	Не ниже версии 3.12
	psycopg2 [2]	Не ниже версии 2.9.6 Для установки библиотеки необходимо предварительно установить пакеты postgresql-devel (yum) и libpq-dev (deb).
ОС	Перечень поддерживаемых ОС приведен в документе «"МойОфис Профессиональный 3". Системные требования»	
Стандартные репозитории ОС	Подключение всех стандартных репозиториев ОС или 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 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)	

[1] Установка или обновление Jinja2 для CentOS выполняется с любого из перечисленных репозиториев OpenStack:

– http://mirror.centos.org/centos/7/cloud/x86_64/openstack-train/

– https://mirror.yandex.ru/centos/7/cloud/x86_64/openstack-train/

[2] Данная библиотека используется для управления конфигурацией сервиса баз данных PostgreSQL при включенном режиме федерации (см. раздел «Режим федерации»).

1.4 Рекомендации по разметке дисков

При разметке дисков необходимо учитывать следующее:

- все рекомендуемые аппаратные требования приведены в разделе «Аппаратные требования»;
- для разных типов установки будут разные требования к выделяемому дисковому пространству;
- для всех серверов рекомендуется оставлять не менее 100 Гбайт на корневой раздел для штатной работы ОС;
- для сервера с ролью `squadus_infra` или при установке в режиме standalone рекомендуется выделять не менее 100 Гбайт на корневой раздел;
- всем серверам рекомендуется выделять отдельный раздел `/srv` для размещения компонентов установки и защиты ОС от переполнения. Дополнительно в разделе можно разместить копии журналов работы компонентов при соответствующей настройке лог-коллектора, но это потребует дополнительного дискового пространства;
- для серверов с ролями `squadus_db` и `squadus_st` рекомендуется выделять независимые диски SSD для серверной части.

1.5 Требования по работе с DNS

1.5.1 Организация работы сервисов разрешения имен

Во время установки производится настройка и запуск локального кеширующего DNS-сервера unbound на серверах с ролью `squadus_apps`. Unbound используется для запросов только внутри продукта и подключается для контейнеров и самих серверов через соответствующие параметры групповых переменных. По умолчанию серверы будут перенастроены на работу через unbound и не будут принимать параметры серверов разрешения имен по DHCP. Поэтому рекомендуется направить unbound на внутренние DNS-серверы компании, если такая необходимость есть. По умолчанию unbound настроен на переадресацию запросов на адреса 8.8.8.8 и 8.8.4.4.

1.5.2 Формирование внешних доменных имен

При установке системы есть возможность указывать метод формирования доменных имен для продукта.

Шаблон, который формирует итоговый вариант всех DNS-записей, принимает на вход значения следующих переменных:

– `squadus_domain` — отображает основной домен, на котором будет работать система;

– `domain_module` — отображает способ формирования доменного имени сервисов ПО.

Пример работы шаблона приведен в таблице 8.

Таблица 8 — Примеры работы шаблона

domain_module	Имя ссылки	squadus_domain	Результат
{service}.{domain}	Auth	example.com	auth.example.com

Изменение принципа формирования доменных имен системы следует выполнять с учетом ограничений (подробнее в разделе «Ограничения по использованию доменного имени»).

1.5.3 Внешние DNS-записи

В таблице 9 приведены необходимые внешние DNS-записи, необходимые для работы системы. Данная таблица сформирована на основании следующих условий:

– для переменной `domain_module` со значением `{service}.{domain}` (т .е. формирование ссылок указывается через точку к указанному домену);

– ansible-переменные `squadus_go_domain`, `squadus_preview_domain`, `jitsi_main_domain`, `scandium_main_domain`, `turnserver_realm` и `squadus_im_domain` (далее — FQDN-шаблоны) не переопределены (подробнее в разделе «Изменение внешних DNS-записей») и имеют значения по умолчанию.

Если выбран другой метод формирования и/или FQDN-шаблоны переопределены, необходимо соотнести его со значениями, представленными в таблице 9.

Используемые DNS-записи могут быть переопределены с помощью FQDN-шаблонов.

Таблица 9 — Сведения о внешних DNS-записях

Имя записи	FQDN-шаблон	Тип записи	Значение	Комментарии
im	squadus_im_ domain	A	IP для приложения	VRRP для squadus_ha при отказоустойчивой установке, IP-адрес сервера с ролью squadus_apps при установке без отказоустойчивости
go	squadus_go_ domain	A	IP для приложения	VRRP для squadus_ha при отказоустойчивой установке, IP-адрес сервера с ролью squadus_apps при установке без отказоустойчивости
meet	jitsi_main_ domain	A	IP сервера с ролью squadus_ha	VRRP для squadus_ha при отказоустойчивой установке, IP-адрес сервера с ролью squadus_apps при установке без отказоустойчивости
scc	scandium_main_ domain	A	IP сервера с ролью squadus_ha	VRRP для squadus_ha при отказоустойчивой установке, IP-адрес с ролью squadus_meet_apps при установке без отказоустойчивости
preview	squadus_preview_ domain	A	IP сервера с ролью squadus_ha	VRRP для squadus_ha при отказоустойчивой установке, IP-адрес сервера с ролью squadus_apps при установке без отказоустойчивости
turn	turnserver_realm	A	IP сервера с ролью squadus_meet_jvb	IP-адрес первого сервера группы squadus_meet_jvb
editor	squadus_wte_ editor_domain	A	IP сервера с ролью squadus_ha	VRRP для squadus_ha при отказоустойчивой установке, IP-адрес сервера с ролью squadus_apps при установке без отказоустойчивости

1.5.4 Изменение внешних DNS-записей

Для изменения названия внешней DNS-записи необходимо добавить переменные для каждого сервиса в файл `main.yml` в директории `squadus_setup` из таблицы 10. По умолчанию переменные принимают значения, указанные в таблице 9. Часть переменных должна иметь одинаковое значение.

Например, переменная `praseodymium_preview_domain` должна наследовать значение переменной `squadus_preview_domain`. Меняя значение одной переменной, необходимо заменить значение второй переменной.

Пример переопределения переменной:

```
squadus_domain: "example.net"
squadus_preview_domain: "preview-new-domain.{{ squadus_domain }}"
praseodymium_preview_domain: "{{ squadus_preview_domain }}"
```

Исходя из указанных значений, переменная `squadus_preview_domain` примет следующий вид:

```
preview-new-domain.example.net
```

Таблица 10 — Переменные, формирующие названия DNS-записей

Переменная	Тип	Описание
jitsi_main_domain	Str	Запись DNS для сервиса meet
praseodymium_ preview_domain	Str	Запись DNS для сервиса praseodymium. По умолчанию запись принимает значение переменной squadus_preview_domain. Обе переменные должны иметь одинаковое значение
scandium_main_ domain	Str	Запись DNS для сервиса scc
squadus_deeplink_url	Str	Запись DNS для сервиса deeplink. По умолчанию запись совпадает с переменной squadus_go_domain. При переопределении переменной необходимо добавить префикс, схему в названии домена — https://
squadus_go_domain	Str	Запись DNS для сервиса go
squadus_im_domain	Str	Запись DNS для сервиса im
squadus_preview_ domain	Str	Запись DNS для сервиса preview
squadus_wte_ editor_domain	Str	Запись DNS для сервиса editor
turnserver_realm	Str	Запись DNS для сервиса turnserver

1.5.5 Внутренние DNS-записи

Все DNS-записи, используемые для работы самой системой 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 `extra_vars.yml` (файл копируется на этапе подготовки к установке, см. раздел «Установка ПО с сервера оператора») или если не использовать unbound и заполнить все необходимые записи во внешнем DNS-сервере. При подобном варианте необходимо создать «A» записи для каждого доменного имени сервера, и записи CNAME для любых поддоменов каждого доменного имени сервера.

Например, «A» запись для `squadus-apps-1.example.net` со значением `192.168.1.2` и CNAME запись для `*.squadus-apps-1.example.net` со значением `squadus-apps-1.example.net`.

1.6 Ограничения

1.6.1 Ограничения по работе с файлом inventory

В файл `hosts.yml` вносятся только доменные имена. Часть логики установщика использует их для формирования доменных имен и адресов сервисов.

1.6.2 Ограничения по работе с подсистемой управления конфигурациями

В соответствии с разделом «Программные требования» на рабочем месте оператора необходимо установить пакеты дополнительного ПО.

В подсистеме управления конфигурациями не должно быть конфигурационных файлов самой подсистемы.

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

Подробнее по ссылке:
https://docs.ansible.com/ansible/latest/reference_appendices/config.html#the-configuration-file

1.6.3 Ограничения по работе с системами виртуализации

Для обеспечения работы ПО поддерживаются следующие системы виртуализации:

- VMware;
- KVM.

1.6.4 Ограничения по использованию доменного имени

При выборе доменного имени и SSL-сертификата следует учитывать, что в настоящее время в качестве основного домена (`domain_name`), на котором будет работать система, могут быть использованы домены только второго уровня.

1.7 Описание архитектуры продукта

Общая архитектурная схема ПО приведена на рисунке 1.

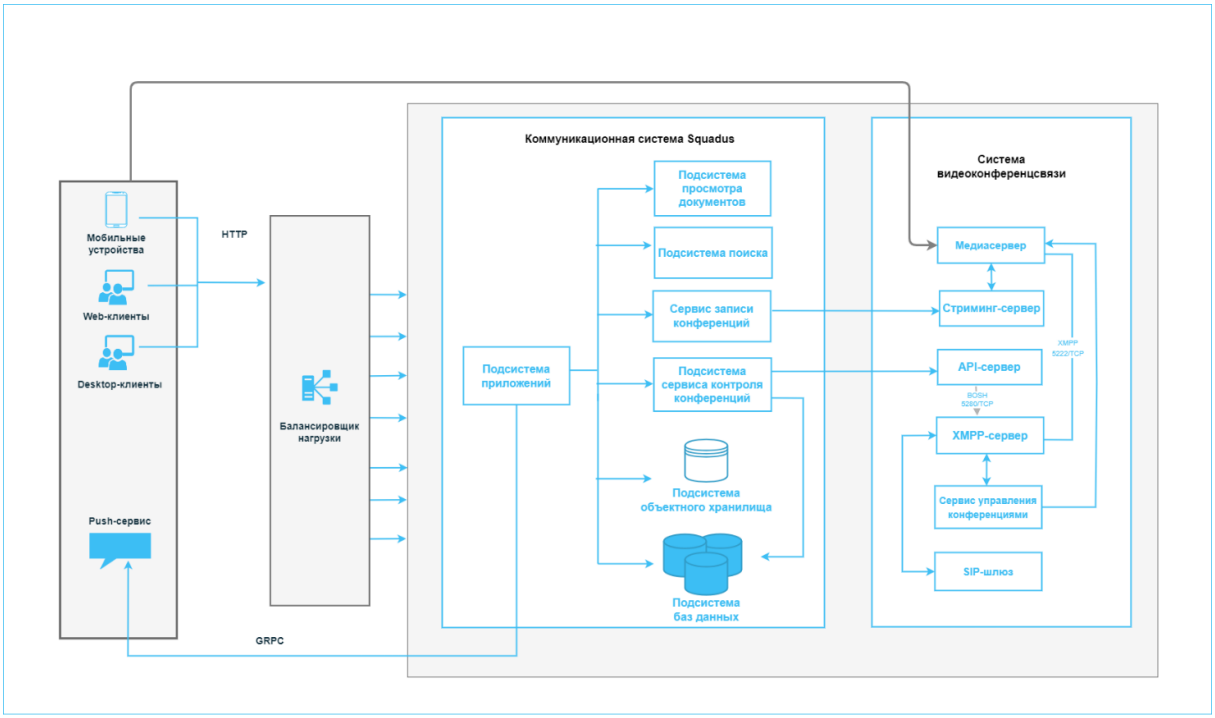


Рисунок 1 — Общая архитектурная схема ПО

1.8 Запуск веб-интерфейса ПО

Пользователи получают доступ к веб-интерфейсу ПО с помощью веб-браузера или настольного приложения. Для запуска ПО с помощью веб-интерфейса необходимо:

1. Открыть браузер при активном сетевом подключении.
2. Ввести в адресную строку браузера адрес вида https://mydomain.ru/.
3. Нажать клавишу **Enter**.
4. Дождаться открытия страницы авторизации ПО.

2 ПЕРВИЧНАЯ УСТАНОВКА

2.1 Состав дистрибутива

В состав дистрибутива ПО входит:

- установщик сервера оператора;
- установщик окружения для проведения установки, включающий все необходимые образы и пакеты;
- файлы tpl (Third-party license);
- руководство по установке ПО.

2.2 Подготовка к установке

2.2.1 Аутентификация в docker registry

По умолчанию в docker registry, устанавливаемом на сервер группы `squadus_infra`, включена аутентификация по логину и паролю для доступа к docker image. Переменные, отвечающие за аутентификацию указаны в таблице 11.

Таблица 11 — Настройка минимальных параметров аутентификации docker registry

Параметр	Описание	Тип	Значение по умолчанию
docker_registry_endpoint	FQDN docker registry	Str	http://{{ docker_image_registry }}
docker_registry_password	Пароль пользователя	Str	oEeUL8kvCkkuxUia8ZaL
docker_registry_username	Имя пользователя	Str	admin

Для отключения аутентификации docker registry необходимо:

- удалить вышеперечисленные переменные из ansible playbook;
- на сервере группы `squadus_infra` удалить файлы `env`, `.htpasswd` выполнив команды:

```
rm -f /srv/docker/registry/conf/env
rm -f /srv/docker/registry/conf/.htpasswd
```

- после выполнения команд перезапустить контейнер docker registry для применения изменений:

```
docker restart docker-registry
```

Для смены пароля, используемого по умолчанию, необходимо:

– изменить пароль в ansible playbook;

– изменить файл `.htpasswd` на сервере группы `squadus_infra` используя утилиту htpasswd.

Пример команды:

```
htpasswd -Bbc /srv/docker/registry/conf/.htpasswd USERNAME PASSWORD
```

где:

`USERNAME` — имя пользователя;

`PASSWORD` — пароль.

2.2.2 Возможные проблемы в работе docker registry

1. Отсутствует DNS-запись для сервера группы `squadus_infra`, при этом запросы доступа к docker registry возвращают ошибку 500.
2. Отсутствует доступ к 5000/tcp порту сервера группы `squadus_infra`.
3. Не совпадают логин и пароль для доступа к docker registry.
4. При смене пароля используется алгоритм хеш-функции, отличный от BCRYPT.

Пример команды:

```
htpasswd -Bbn admin oEeUL8kvCkkuxUia8ZaL
```

2.2.3 Описание общих ролей подсистемы Ansible

Общие роли подсистемы управления конфигурациями для преднастройки серверов перед установкой описаны в таблице 12.

Таблица 12 — Описание общих ролей Ansible

Наименование роли	Описание
nct.system.authorized_keys	Добавляет указанные ssh-ключи для выбранных пользователей на серверы группы play_hosts
nct.system.hostname	Устанавливает hostname для выбранных серверов группы play_hosts
nct.system.selinux	Проверяет режим работы SELinux и переключает его в режим «enforcing»
nct.system.yum	Настраивает пакетный менеджер (yum), обновляет все пакеты до последней актуальной версии в подключенных репозиториях, за исключением kernel*, docker-ce*, container*
nct.system.package_tools	Добавляет требуемые пакеты для работы Хранилища в целевую ОС
nct.system.locale	Устанавливает параметры locale на серверы группы play_hosts

Наименование роли	Описание
nct.system.timezone	Устанавливает часовой пояс на серверы группы play_hosts
nct.system.sshd	Производит настройку службы sshd
nct.system.chrony	Устанавливает и настраивает службу синхронизации времени chronyd
nct.system.timesyncd	Устанавливает и настраивает службу синхронизации времени
nct.system.sysctl	Устанавливает требуемые параметры ядра на серверы группы play_hosts
nct.system.limits	Настраивает параметры лимитов на серверы группы play_hosts
nct.system.kernel_ml	Устанавливает пакет kernel_ml последнего доступного ядра
nct.system.rsyslog	Устанавливает и настраивает сервис сбора журналов
nct.system.docker	Устанавливает и настраивает docker, подключает к docker registry
nct.system.iptables	Устанавливает и настраивает службы межсетевого экрана с параметрами, требуемыми для конкретной роли
nct.system.unbound	Устанавливает и настраивает кеширующий DNS-сервер
nct.system.resolv	Настраивает правила разрешения DNS-серверов
nct.system.python3	Устанавливает python3 с необходимыми зависимостями

2.2.4 Роли, используемые для установки ПО

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

Таблица 13 — Описание ролей, используемых для установки ПО Squadus

Наименование роли	Описание
nct.tools.remover	Удаляет сервисы в существующей системе
nct.redis.redis	Устанавливает и настраивает Redis
nct.redis.cluster	Настраивает cluster Redis
nct.monitoring.redis_exporter	Устанавливает и настраивает redis_exporter
nct.mongodb.mongodb	Устанавливает и настраивает базу данных MongoDB
nct.mongodb.backup	Устанавливает и настраивает резервное копирование базы данных MongoDB
nct.mongodb.mongodb_backup	Устанавливает и настраивает резервное копирование базы данных MongoDB с использованием ПО Percona Backup for MongoDB
nct.monitoring.mongodb_exporter	Устанавливает и настраивает mongodb_exporter для мониторинга mongodb
nct.monitoring.postgresql_exporter	Устанавливает и настраивает postgresql_exporter для мониторинга postgresql
nct.zookeeper.zookeeper	Устанавливает и настраивает zookeeper
nct.minio.minio	Устанавливает и настраивает S3-совместимое хранилище MinIO

Наименование роли	Описание
nct.minio.minio_backup	Устанавливает и настраивает систему резервного копирования S3-совместимого хранилища MinIO
nct.co.jod	Устанавливает и настраивает компонент jod системы предпросмотра документов
nct.co.pregen	Устанавливает и настраивает компонент pregen системы предпросмотра документов
nct.co.cvm	Устанавливает и настраивает компонент cvm системы предпросмотра документов
nct.nginx.nginx	Устанавливает и настраивает nginx
nct.logging.logrotate	Настраивает ротацию логов
nct.logging.syslog_ng	Устанавливает и настраивает централизованный сбор логов сервисов на ВМ группы squadus_infra
nct.ha.envoy	Устанавливает и настраивает envoy proxy
nct.squadus.squadus	Устанавливает и настраивает основной компонент squadus
nct.squadus.squadus_jiratrigger	Устанавливает и настраивает сервис интеграции suqadus_jiratrigger с Jira
nct.squadus.bohrium	Устанавливает и настраивает компонент bohrium — сервис, реализующий сохранение и удаление файлов записей встреч (DAL)
nct.squadus.cobalt	Устанавливает и настраивает компонент cobalt — сервис для работы с настройками
nct.squadus.lithium	Устанавливает и настраивает компонент lithium — сервис rate-limiter
nct.squadus.manganum	Устанавливает и настраивает компонент manganum — сервис DAL (data access layer)
nct.squadus.tennessine	Устанавливает и настраивает компонент tennessine — сервис статусов пользователей
nct.squadus.zinc	Устанавливает и настраивает компонент zinc — сервис FAL (file access layer)
nct.squadus.synapse	Устанавливает и настраивает компонент synapse — сервис федерации
nct.mail.postfix	Устанавливает и настраивает сервер отправки и пересылки почты
nct.mail.opendkim	Устанавливает и настраивает систему цифровой подписи email сообщений
nct.monitoring.alertmanager	Устанавливает и настраивает систему оповещений alertmanager
nct.monitoring.blackbox_exporter	Устанавливает и настраивает blackbox_exporter для мониторинга срока действия сертификатов, доступности веб-серверов
nct.monitoring.cadvisor	Устанавливает и настраивает cadvisor для сбора метрик docker контейнеров
nct.monitoring.grafana	Устанавливает и настраивает систему мониторинга и визуализации grafana

Наименование роли	Описание
nct.monitoring.jitsi_exporter	Устанавливает и настраивает сервис сбора метрик для подсистемы видеоконференций
nct.monitoring.node_exporter	Устанавливает и настраивает сервис сбора метрик node_exporter
nct.monitoring.prometheus	Устанавливает и настраивает систему мониторинга и визуализации prometheus
nct.monitoring.pushgateway	Устанавливает и настраивает систему мониторинга и визуализации для сбора custom метрик pushgateway
nct.postgresql.postgresql	Устанавливает и настраивает базу данных PostgreSQL
nct.postgresql.backup	Устанавливает и настраивает резервное копирование базы данных PostgreSQL
nct.search.squash	Устанавливает и настраивает сервис поиска squash_search, squadus_scatter
nct.search.chatpal	Устанавливает и настраивает систему поиска для ПО
nct.squadus.jiratrigger	Устанавливает сервис получения информации из JIRA
nct.jitsi.scandium	Устанавливает и настраивает сервис управления конференциями
nct.jitsi.vosk	Устанавливает и настраивает сервис транскрибации
nct.ha.keepalived	Устанавливает и настраивает сервис keepalived
nct.ha.haproxy	Устанавливает и настраивает proxy haproxy
nct.jitsi.excalidraw	Устанавливает и настраивает сервис для доски совместного рисования в видеоконференциях
nct.jitsi.jicofo	Устанавливает и настраивает различные компоненты подсистемы видеоконференций
nct.jitsi.jitsiweb	Устанавливает и настраивает различные компоненты подсистемы видеоконференций
nct.jitsi.jigasi	Устанавливает и настраивает сервис jigasi
nct.jitsi.jvb	Устанавливает и настраивает сервис jitsi videobridge
nct.jitsi.jibri	Устанавливает и настраивает сервис записи видеоконференций
nct.jitsi.turnserver	Устанавливает и настраивает STUN/TURN сервер для конференций

2.2.5 Подготовка инфраструктуры установки

Для подготовки инфраструктуры установки необходимо выполнить следующие действия:

1. Установить хранилища образов Docker (docker_registry).
2. Установить подсистемы управления конфигурациями (Ansible).

Подробные сведения о выполнении указанных действий приведены в разделах: «Установка подсистемы управления конфигурациями (Ansible)» и «Установка хранилища образов Docker».

2.2.6 Установка хранилища образов Docker

Установка производится на сервере с ролью `squadus_infra`. Перед началом установки необходимо проверить, что вход выполнен под пользователем root.

Этапы установки:

1. Скопировать файлы squadus_infra_<RELEASE>.run на сервер.
2. Запустить скрипт установки:

```
bash squadus_infra_<RELEASE>.run
```

3. Согласиться на продолжение установки, нажав на клавишу «Y».

```
root@squadus_infra ~]# bash ./squadus_infra_<RELEASE>.run
Verifying archiveintegrity...100%                All good.
Uncompressing Squadus Infrastructure Node Package <RELEASE>100%
#
# Copyright (c) New Cloud Technologies, Ltd., 2013-2024 #
# You can not use the contents of the file in any way without New Cloud Technologies, Ltd. written permission.
# To obtain such a permit, you should contact New Cloud Technologies, Ltd. at http://ncloudtech.com/contact.html
# Welcome to Squadus Infrastructure Installer version <RELEASE>

This script is meant to be used on Infrastructure Server (see manualfor default)
Do you want to continue? [y/N] Y
Check if the script is run with superuser privileges        [ OK ]
Make sure that the operating systemis compatible            [ OK ]
Ensurethat yum-utils is installed                           [ OK ]
```


Администратору отобразится следующее (список отображения может меняться в зависимости от выбранной ОС):

```
      Ensure that docker-ce repository is availiable              [ OK ]

      Ensure that docker is installed                             [ OK ]
      Ensure that jq package is installed                         [ OK ]
      Ensure that docker dir exists                               [ OK ]
      Ensure that docker daemon config exists                     [ OK ]
      Check if docker daemon needs to be restarted                [ OK ]
      Ensure that docker is started                               [ OK ]
      Ensure that docker is enabled                               [ OK ]
      Check if docker is availiable                               [ OK ]
      Ensure that registry image is available                     [ OK ]
      Check if container with registry is available               [CHANGE]
      Ensure that registry configuration directory exists         [ OK ]
      Ensure that docker-registry env file exists                 [ OK ]
      Check if old registry data directory exists                 [CHANGE]
      Ensure that registry data directory exists                  [CHANGE]
      Ensure that container with registry is available            [CHANGE]
      Wait for docker-registry to start                           [ OK ]
      Ensure that docker-registry is running                      [ OK ]
      Extracting registry archive...                              [ OK ]
      Remove dangling and outdated images                         [ OK ]
```

После этого установка хранилища образов Docker (docker_registry) будет завершена.

2.2.7 Установка подсистемы управления конфигурациями (Ansible)

Установка производится на рабочем месте оператора. Необходимо проверить соблюдение следующих условий:

1. Вход выполнен под пользователем root или под пользователем с sudo-привилегиями на пакетный менеджер.

2. Машина, на которой выполняется установка, соответствует требованиям, указанным в разделе «Системные требования».

3. С выбранного сервера есть возможность доступа по SSH-протоколу к другим серверам, на которых выполняется установка.

4. Подсистема управления конфигурациями Ansible установлена, другие конфигурационные файлы Ansible не присутствуют в ОС.

Этапы установки:

1. Скопировать файл `squadus_ansible_bin_<RELEASE>.run` в домашнюю директорию пользователя root.
2. Запустить файл squadus_ansible_bin_<RELEASE>.run с помощью команды:

```
      bash squadus_ansible_bin_<RELEASE>.run
```


3. Согласиться на продолжение установки, нажать на клавишу "Y".

```
[root@squadus_infra ~]# bash ./squadus_ansible_bin_<RELEASE>.run Verifying archiveintegrity...100% All good.
Uncompressing SQUADUSAnsible Package <RELEASE>100%
#
# Copyright (c) New Cloud Technologies, Ltd., 2013-2024 #
# You can not use the contents of the file in any way without New Cloud Technologies, Ltd. written permission.
# To obtain such a permit, you should contact New Cloud Technologies, Ltd. at http://ncloudtech.com/contact.html
# Welcome to Squadus AnsibleInstaller version <RELEASE>
This script is meant to be used on operator workstation
Do you want to continue? [y/N]Y
Install ucs-storm[ OK ]
Ensure that python-netaddr is installed                [ OK ]
Ensure that python2-jmespath is installed              [ OK ]
Ensure that version directory is present               [ OK ]
Ensure that version <RELEASE> is present               [ OK ]
Set <RELEASE> as latest                                [ OK ]
Create roles symlink                                   [ OK ]
Create collections symlink                             [ OK ]
Create contrib symlink                                 [ OK ]
```

Администратору отобразится следующее (список отображения может меняться в зависимости от выбранной ОС):

```
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 ]
Create symlink to module:ucs-storm                     [ OK ]
```

После этого установка подсистемы управления конфигурациями будет завершена.

2.2.8 Установка ПО сервера оператора

Этапы установки:

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

```
root@squadus_infra~]# cd ~/install_squadus
```

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

```
[root@squadus_infra~]# cp contrib/squadus/ansible.cfg
```

3. Выбрать наиболее подходящий тип установки:

   – standalone (установка без поддержки отказоустойчивости);

   – cluster (установка с поддержкой отказоустойчивости).

Предзаполненные файлы с примерами заполнения параметров установки располагаются по пути

```
~/install_squadus/contrib/squadus/[cluster | standalone]_hosts.yml
```


4. Скопировать заготовку файла `hosts.yml` в директорию `~/install_squadus/`:

```
cp contrib/squadus/<тип установки>_hosts.yml ./hosts.yml
```

5. Заполнить заготовку файла `hosts.yml`, указывая FQDN, по которым серверы доступны. Следует учитывать, что роли `squadus_ha` и `squadus_apps` не должны совпадать, если хостов в `squadus_apps` более одного.

6. Перенести заготовку файлов параметров `group_vars` с помощью команды:

```
cp -r contrib/squadus/group_vars/squadus_setup ./group_vars/
```

При необходимости можно переименовать `squadus_setup` на произвольное имя, но тогда необходимо изменить таким же образом имя группы `squadus_setup` в файле inventory.

8. Открыть файл `main.yml` из каталога размещения и отредактировать значения параметров по комментариям. Примеры параметров для минимальной настройки можно найти в самих файлах, а также в разделе «Настройка минимальных параметров установки».

9. Скопировать ssl-ключи для внешнего домена в каталог `certificates`. Подробнее см. в разделе «Размещение ssl-сертификатов для шифрования».

2.2.9 Размещение SSL-сертификатов для шифрования

Требования к сертификатам:

1. При создании сертификата следует учитывать, что SAN должен содержать необходимые доменные имена (см. полный список в разделе «Внешние DNS-записи»).
2. Допускается использование Wildcard SSL-сертификата.
3. Для корректной работы сертификат должен быть подписан авторизованным сертификационным центром.
4. Сертификаты должны обязательно соответствовать формату PEM и содержать «BEGIN CERTIFICATE». Сертификаты OpenSSL формата PEM с заголовком «BEGIN TRUSTEDCERTIFICATE» не поддерживаются.

Порядок размещения сертификатов:

1. Разместить сертификат внешнего домена:

```
[root@squadus_infra ~]# vim certificates/server.crt
```

2. Разместить ключ внешнего домена:

```
[root@squadus_infra ~]# vim certificates/server.nopass.key
```

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

```
[root@squadus_infra ~]# vim certificates/ca.pem
```

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

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

Для создания самоподписанного (selfsigned) Wildcard SSL сертификата можно воспользоваться скриптом из поставки. После запуска приведенной команды генерируются все необходимые сертификаты и ключи для домена `example.net`, а также автоматически копируются в директорию `certificates`.

```
bash contrib/scripts/gen_self_signed_cert.sh example.net
```

2.2.10 Обновление SSL-сертификатов

Для обновления ранее установленных сертификатов необходимо разместить новые сертификаты в директории `certificates` согласно порядку размещения, описанном в разделе «Размещение ssl-сертификатов для шифрования», заменив старые файлы на новые.

После выполнения шагов, описанных в разделе «Размещение ssl-сертификатов для шифрования», необходимо выполнить установку новых сертификатов с помощью команды:

```
ansible-playbook playbooks/squadus/ingress.yml
```

2.2.11 Настройка основных параметров установки

2.2.11.1 Настройка минимальных параметров установки

Перечень переменных, необходимых для установки, приведен в таблице 14.

В целях обеспечения безопасности все пароли и ключи, приведенные в таблице 14 и далее, должны быть заменены на новые.

Таблица 14 — Настройка минимальных параметров установки

Параметр	Описание	Тип	Значение по умолчанию
ansible_user	Имя пользователя, от которого выполняется установка. Пользователь должен иметь доступ по ssh к хостам системы	Str	-
jibri_recorder_password	Пароль, используемый в подсистеме записи конференций	Str	-
jibri_auth_password	Пароль для внутренней авторизации сервиса записи конференций	Str	-
jicofo_auth_password	Пароль для аутентификации jicofo (компонент видеоконференций)	Str	-

Параметр	Описание	Тип	Значение по умолчанию
jicofo_component_secret	Внутренний пароль для работы jicofo (компонент видеоконференций)	Str	-
jitsi_jwt_secret	JWT-пароль для работы jitsi (компонент видеоконференций)	Str	-
jitsi_jwt_app_id	Идентификатор JWT приложения	Str	-
jvb_auth_password	Аутентификационный пароль для работы jvb (компонент видеоконференций)	Str	-
jvb_component_secret	Внутренний пароль для работы jvb (компонент видеоконференций)	Str	-
manganum_mongodb_password	Пароль для базы данных, используемой компонентом manganum. Должен ссылаться на пароль сервиса squadus	Str	-
minio_access_key	Ключ доступа к хранилищу S3	Str	-
minio_secret_key	Секретный ключ для доступа к хранилищу S3	Str	-
mongodb_root_password	Пароль привилегированного пользователя в базе данных MongoDB	Str	-
mongodb_secured_key	Секретный ключ для установки MongoDB	Str	-
redis_password	Пароль к базе данных Redis	Str	-
squadus_domain	Указывается внешнее доменное имя системы для доступа к ПО	Str	-
squadus_mongodb_password	Пароль для базы данных, используемой сервисом squadus	Str	-
jitsi_use_keepalived	Включение и настройка сервиса keepalived для работы в кластерном режиме	Bool	-
tls_ca_filename	Имя файла цепочки промежуточных сертификатов CA для внешнего домена	Str	ca.crt
tls_cert_filename	Имя файла сертификата для внешнего домена	Str	server.crt
tls_key_filename	Имя файла ключа для внешнего домена	Str	server.nopass.key
squadus_use_keepalived	Включение и настройка сервиса keepalived для работы в кластерном режиме	Bool	-
scandium_mongodb_password	Пароль для базы данных, используемой сервисом управления конференциями	Str	-

Параметр	Описание	Тип	Значение по умолчанию
squadus_jwt_secret	JWT секрет для подключения ПО для сервиса управления конференциями	Str	Vagi2uk2CheCahbohphe
tennessine_mongodb_password	Пароль для базы данных, используемой компонентом tennessine. Должен ссылаться на пароль сервиса squadus	Str	-
zinc_mongodb_password	Пароль для базы данных, используемой компонентом zinc. Должен ссылаться на пароль сервиса squadus	Str	-
turnserver_cli_password	Пароль для доступа к администраторскому командному интерфейсу (CLI) - указано в файле дистрибутива readme	Str	RcBaN5mKibWe25h6NWiN
turnserver_secret	Внутренний пароль для работы API	Str	oogieyahneiBienei8ey

Пример корректно настроенных параметров:

```
ansible_user: "root"
jibri_recorder_password: "eChuoNgae0Ohz1ihahmoichu4heeshu4"
jibri_auth_password: "aCeephaobahjeexeWei0Aa8ceeboegiu"
jicofo_auth_password: "concentrationcompensationgivesreed"
jicofo_component_secret: "ieH1iedaejai0ob9piC1faeGhiel2ahM"

jitsi_jwt_secret: "aishi8ceYi6peid1Yeemaawahb2ve7we"
jitsi_jwt_app_id: "meet"
jitsi_main_domain: "meet.example.net"
jitsi_use_keepalived: false

jvb_auth_password: "solelyafternoonattorneyssomewhere"
jvb_component_secret: "moopheh1aixohcaequai5go4Awee4ou7"

manganum_mongodb_password: "{{ squadus_mongodb_password }}"

minio_access_key: "thi1shogeuThu0sheeShooqueukur8Ae"
minio_secret_key: "neecien8Gah0iudoh6Ooloong5oopaem"
mongodb_root_password: "processionmerryrapidmessage"
mongodb_secured_key: "elephantwisdomexceptionuse"

redis_password: "rakeproposebowpupilvirtue"

squadus_jitsi_enabled: true
squadus_domain: "example.net"
squadus_mongodb_password: "illegalongoingsurvivedesignation"
squadus_use_keepalived: false
scandium_mongodb_password: "chopohYuicoo2moo2uuphev2iechae4E"
scandium_squadus_jwt_secret: "chaevieT0oiz3ifoocev3eevainei4om"
```

```
      tennessine_mongodb_password: "{{ squadus_mongodb_password }}"

      tls_ca_filename: "ca.pem"
      tls_cert_filename: "server.crt"
      tls_key_filename: "server.nopass.key"
      turnserver_cli_password: "RcBaN5mKibWe25h6NWiN"
      turnserver_secret: "oogieyahneiBienei8ey"

      zinc_mongodb_password: "{{ squadus_mongodb_password }}"
```

2.2.11.2 Установка системы для работы более 1000 пользователей

При установке ПО для использования 1000 и более пользователей необходимо с помощью текстового редактора открыть файл, расположенный в следующей директории `inventory/group_vars/squadus/main.yml` и вручную добавить с новой строки переменную:

```
      tennessine_status_debounce_time: 7000
```

2.2.12 Настройка дополнительных параметров установки

Настройка дополнительных параметров производится в файле `extra_vars.yml`. Данные параметры изменять не обязательно. Настройка дополнительных параметров установки приведена в таблице 15.

Таблица 15 — Настройка дополнительных параметров установки

Параметр	Описание	Тип	Значение по умолчанию
excalidraw_whiteboard_enabled	Включение виртуальной доски для рисования в видеоконференциях	Bool	false
jitsi_sip_enabled	Установка и настройка сервиса для интеграции со сторонней SIP системой	Bool	false
jigasi_sip_server	Адрес SIP сервера, с которым проводится интеграция	Str	-
jigasi_sip_port	Порт сервера SIP	Int	-
jigasi_sip_transport	Транспортный протокол, используемый сервером SIP, с которым проводится интеграция	Str	-
jigasi_auth_password	Внутренний пароль для сервиса jigasi	Str	-
jigasi_sip_password	Пароль для доступа к SIP серверу, с которым проводится интеграция	Str	-
jitsi_readonly_username_enabled	Разрешает или запрещает изменение отображаемого имени пользователя при подключении к конференциям	Bool	false
ntp_servers	Список NTP серверов для синхронизации времени	List	- "centos.pool.ntp.org"
squadus_enable_unbound	Устанавливает внутренний кеширующий DNS-сервер	Bool	true

Параметр	Описание	Тип	Значение по умолчанию
unbound_access_ control	Словарь, описывающий сетки, которые имеют доступ к DNS-серверу	Dict	-
unbound_forward_ addresses	Список адресов серверов, к которым DNS-сервер будет перенаправлять запросы, если не может ответить самостоятельно	List	- “8.8.8.8” - “8.8.4.4”

Пример корректно настроенных параметров:

```
excalidraw_whiteboard_enabled: true

jitsi_sip_enabled: true
jigasi_sip_server: "10.1.1.51"
jigasi_sip_port: 5160
jigasi_sip_transport: "udp"
jigasi_auth_password: "wepeez3euQueik7pohke2cixohciePhu"
jigasi_sip_password: "Uzaingai6iuwo8aesuw9aThe6Jie7Ru5"

squadus_enable_unbound: true

unbound_access_control: network1:"192.168.1.0/24"

unbound_forward_addresses:
-"8.8.8.8"
-"9.9.9.9"

ntp_servers:
  - "ntp01.your-domain.ru"
  - "ntp02.your-domain.ru"
  - "ntp03.your-domain.ru"
```

2.2.13 Настройка межсетевого экранирования

Во время установки происходит настройка межсетевого экрана внутри контура системы. Необходимо обеспечить дополнительную защиту системы с внешней стороны системы (по отношению к контуру).

Во внешний контур должны быть доступны 80/tcp и 443/tcp порты с IP-адресов виртуальных серверов с ролью `squadus_ha`. Следует разрешить порты 4096/udp и 10000/udp на `squadus_meet_jvb`. А также порты 3478[tcp|udp], 5349[tcp|udp] и 49152:65535/udp на первый сервер группы `squadus_meet_jvb` для работы turnserver.

Остальные порты должны быть запрещены.

2.2.14 Настройка службы синхронизации времени NTP

Настройка синхронизации времени выполняется автоматически при применении `playbook-a playbooks/common.yml` в соответствии с разделом «Настройка дополнительных параметров установки». При необходимости пул NTP серверов можно переопределить с помощью переменной `ntp_servers`.

Необходимо учитывать, что корректная работа системы синхронизации времени крайне важна для корректной работы ПО. Стоит убедиться, что используемые NTP серверы доступны по UDP-порту 123 со всех узлов установки.

2.3 Запуск установки

Установку можно выполнить двумя способами:

1. Для установки потребуется выполнить следующие действия:

– Запустить команду на подготовку серверов к установке:

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

После запуска этой команды будут запущены роли, указанные в разделе «Описание общих ролей подсистемы Ansible».

– Запустить команду на установку ПО:

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

После этого запускаются роли, указанные в разделе «Роли, используемые для установки ПО».

2. Запустить объединенный playbook, который выполнит подготовку серверов и установку ПО:

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

2.4 Проверка корректности установки

Для проверки корректности установки необходимо выполнить следующие действия:

1. Запустить приложение ПО.
2. Использовать для входа учетные данные пользователя или администратора.
3. Выбрать контакт из списка и отправить ему сообщение.
4. Если сообщение успешно отправилось и было прочитано пользователем — система настроена корректно.

2.5 Запуск нескольких экземпляров ПО

При установке в отказоустойчивой конфигурации, возможен запуск нескольких экземпляров ПО Squadus на каждой из нод группы `squadus_apps`.

Для этого необходимо выполнение следующих условий:

1. Отсутствие совмещения ролей на серверах группы `squadus_apps` с ролями `squadus_db`, `squadus_meet`, `squadus_st`.
2. Удовлетворение характеристик серверов следующим параметрам:
   – количество vCPU — больше 4;
   – объем RAM — больше 8 Гбайт.

Количество запускаемых экземпляров регулируется переменной squadus_pool_size и рассчитывается автоматически по формуле squadus_pool_size = vCPUS – 4, если vCPUS > 4 и RAM > 8 Гбайт. В ином случае запускается один экземпляр.

Количество запускаемых экземпляров на серверах можно переопределить в inventory-файле (`hosts.yml`), добавив переменную для нужного сервера или группы серверов:

```
squadus_apps:
  hosts:
    squadus-apps-1.example.net:
      squadus_pool_size: 2
        squadus-apps-[2:3].example.net:
```

Необходимо обеспечить минимальные аппаратные требования для одного экземпляра 1 vCPU и 2 Гбайт RAM.

После изменения переменной необходимо применить изменения для сервисов `squadus`, `nginx` и `prometheus`.

2.6 Настройка push-уведомлений в режиме удаленного вызова процедур GRPC

Настройка push-уведомлений выполняется в ручном или автоматическом режиме.

Автоматическая настройка производится с использованием переменных из файла `group_vars/squadus_setup/main.yml`. Все параметры передаются в environment контейнера сервиса `squadus` и при перезапуске контейнера будут применены повторно.

Автоматическая настройка невозможна без повторного запуска `ansible-playbook` с последующим перезапуском контейнеров `squadus`. На время перезапуска сервисы будут недоступны.

Ручная настройка предполагает ввод параметров подключения через административную панель. При настройке перезапуск контейнеров не потребуется.

Для настройки push-proxy администратору стенда `squadus` будут переданы следующие данные:

– файл сертификата CA центра;
– файл сертификата клиента подписанный CA центром;
– файл с ключом для сертификата клиента;
– ID клиента;
– пароль клиента;
– список серверов для подключения по GRPC с указанием FQDN серверов, и порт подключения (по умолчанию используется порт 3031/tcp).

ⓘ	Список серверов для подключения задан по умолчанию в переменной `myoffice_push_grpc_proxy_endpoints`

2.6.1 Ручная настройка

Перед началом настройки push-proxy необходимо убедиться, что пользователь обладает правами администратора.

Для включения настройки следует открыть административную панель и перейти в раздел **Администрирование** > **Push уведомления**.

В соответствии с рисунком 2 необходимо установить переключатель в правое положение **Включить**.

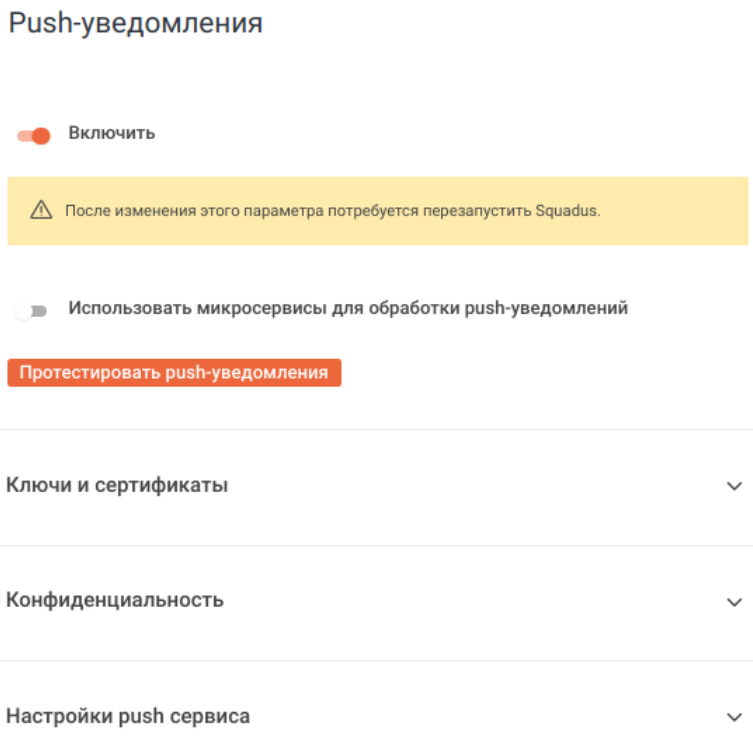


Рисунок 2 — Окно подключения push-уведомлений

После включения сервиса в списке параметров необходимо выбрать пункт **Настройки push сервиса**.

В открывшемся окне (см. Рисунок 3) следует указать значения для полей, описанных в таблице 16.

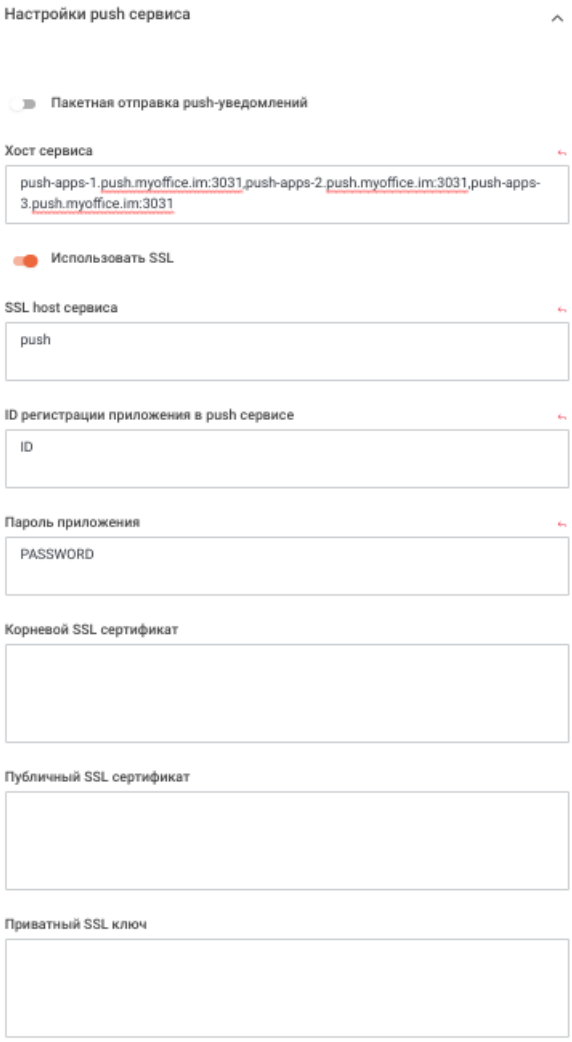


Рисунок 3 — Окно настройки push-сервиса

Таблица 16 — Настройка полей push-сервиса

Наименование поля	Примеры значений	Описание
Хост сервиса	push-apps-1.push.myoffice.im:3031, push-apps-2.push.myoffice.im:3031, push-apps-3.push.myoffice.im:3031	Список серверов push-proxy с указанием порта через двоеточие, разделенный запятой для отправки push-уведомлений
SSL-хост сервиса	push	Subject alternative name сертификата
ID регистрации приложения в push сервисе	-	ID для подключения
Пароль приложения	-	Пароль
Корневой SSL сертификат	-	Содержимое файла myoffice_ca.pem

Наименование поля	Примеры значений	Описание
Публичный SSL сертификат	-	Содержимое файла mycompany_client.crt
Приватный SSL ключ	-	Содержимое файла mycompany_client.nopass.key

После заполнения указанных полей необходимо нажать кнопку **Сохранить** в правом верхнем углу страницы.

2.6.2 Автоматическая настройка

Для настройки push-уведомлений в автоматическом режиме необходимо выполнить следующие действия:

1. Раскомментировать и изменить переменные, указанные в таблице 17.

Таблица 17 — Переменные для настройки push-уведомлений

Наименование переменной	Описание
myoffice_push_proxy_grpc_enabled	Включает режим проксирования к push-серверам через удаленный вызов процедур gRPC (значение true)
myoffice_push_grpc_proxy_endpoints	Список push-серверов
myoffice_push_proxy_tls_ca_filename	Сертификат центра сертификации, подписавшего используемые сертификаты
myoffice_push_proxy_tls_cert_filename	Клиентский сертификат для подключения к сервису push-уведомлений
myoffice_push_proxy_tls_key_filename	Ключ сертификата

2. Разместить полученные файлы в рабочей директории дистрибутива `certificates`.

3. Повторно запустить установку сервиса `squadus` с помощью команды:

```
[root@squadus_infra~]# ansible-playbook –i hosts.yml playbooks/main.yml\
 –l squadus_apps –t squadus
```

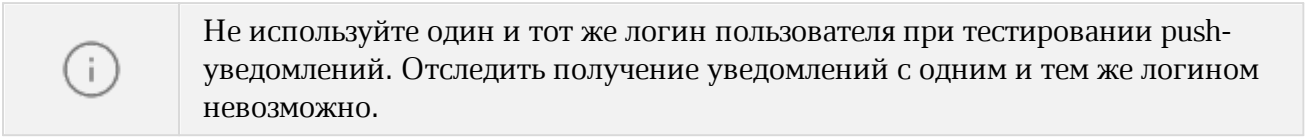
4. После завершения работы Ansible в административной панели ПО проверить применение настроек в соответствии с разделом «Ручная настройка».

2.6.3 Проверка работы сервиса

Для проверки необходимо выполнить следующие операции:

– перейти в чат с пользователем и совершить звонок;

– проверить телефон пользователя во время звонка на наличие push-уведомления.



2.6.4 Устранение неполадок

При обновлении настроек push-proxy необходимо сохранить настройки и проверить работу push-уведомлений.

При отсутствии push-уведомлений следует перезапустить один из контейнеров группы `squadus_apps` для обновления и применения настроек.

Пример команды:

```
docker restart -t 120 squadus-1
```

2.7 Резервное копирование данных

2.7.1 Резервное копирование MongoDB

Резервное копирование MongoDB выполняется двумя способами:

1. При использовании обычного резервного копирования база сохраняется, без возможности сохранения масштабированной MongoDB.

2. При использовании ПО Percona Backup for MongoDB резервная копия сохраняется для масштабированной MongoDB.

2.7.1.1 Резервное копирование без дополнительного ПО

Данный тип резервного копирования предназначен для создания резервной копии установки в режиме standalone и с использованием Replica Set.

Резервное копирование баз данных MongoDB настраивается в автоматическом режиме ролью `nct.mongodb.backup`. Роль создает расписание cron в директории `/etc/cron.d/` с ежедневным запуском создания резервной копии баз данных в 01.00 AM. Для изменения расписания необходимо добавить переменные в файл `group_vars/squadus/main.yml`, изменив значения переменных. Переменные для настройки резервного копирования приведены в таблице 18.

Таблица 18 — Настройка параметров расписания резервного копирования MongoDB

Название переменной	Описание	Тип	Значение по умолчанию
mongodb_backup_cron_time_day	Устанавливает день	Str	"*"
mongodb_backup_cron_time_hour	Устанавливает час	Int	1
mongodb_backup_cron_time_minute	Устанавливает минуту	Int	0
mongodb_backup_cron_time_month	Устанавливает месяц	Str	"*"
mongodb_backup_cron_time_weekday	Устанавливает день недели	Str	"*"
mongodb_backup_keep_days	Устанавливает количество сохраняемых архивов при ротации после успешного выполнения задания	Int	7
docker_backup_dir	Директория для сохранения резервных копий	Str	"/srv/backups"

Пример корректно настроенных параметров:

```
mongodb_backup_cron_time_day: "*"
mongodb_backup_cron_time_hour: 1
mongodb_backup_cron_time_minute: 0
mongodb_backup_cron_time_month: "*"
mongodb_backup_cron_time_weekday: "*"
mongodb_backup_keep_days: 7
docker_backup_dir: "/srv/backups"
```

2.7.1.2 Использование ПО Percona Backup for MongoDB

Для выполнения резервного копирования с помощью Percona Backup for MongoDB в системе должен присутствовать отдельный сервер с ролью squadus_infra.

Percona Backup for MongoDB (далее — PBM) позволяет создавать резервные копии MongoDB, работающей в режиме шардирования, или в режиме ReplicaSet (этот режим подразумевает работу без остановки сервисов MongoDB).

Для создания резервных копий MongoDB, работающей в режиме шардирования ReplicaSet без остановки сервисов MongoDB используется ПО Percona Backup for MongoDB (PBM). Ограничением является работа MongoDB без ReplicaSet.

Для создания резервной копии MongoDB с использованием ПО PBM необходимо на каждую ноду MongoDB установить агент ПО PBM с подключением к MongoDB. Логин и пароль пользователя для создания резервной копии добавляется в MongoDB автоматически при установке MongoDB.

Установленный агент ПО PBM подключается к каждой ноде MongoDB с указанием:

– ReplicaSet;

– порта подключения;

– префикса и FQDN ноды MongoDB.

При работе MongoDB в шардированном режиме у каждой ноды сервера будет отличаться порт, префикс, название ReplicaSet ноды.

Резервные копии, созданные ПО PBM, сохраняются в хранилище MinIO, устанавливаемое на виртуальную машину группы `squadus_infra`. С каждой ноды MongoDB для агента ПО PBM должен быть предоставлен доступ к виртуальной машине `squadus_infra` по 9000/tcp порту.

Если ПО PBM используется в системе после развертывания, необходимо выполнить команду:

```
anbile-playbook -i hosts.yml playbooks/squadus/infra.yml -t mongodb
```

Изменения будут применены, а также добавятся все необходимые реквизиты для создания резервных копий.

Для включения резервного копирования необходимо в файле `inventory/group_vars/squadus/main.yml` изменить переменную `mongodb_backup_mongodb_pbm_enabled`, присвоив значение `true`.

Установка ПО PBM выполняется с помощью команды:

```
ansible-playbook -i hosts.yml playbooks/squadus/backup.yml -t
mongodb_backup
```

Проверка резервного копирования:

1. Создание резервных копий запускается автоматически по расписанию сервиса cron, настраиваемом аналогично таблице 18. Для ручного запуска следует выполнить скрипт:

```
/srv/docker/mongodb_backup_management/backup_scripts/ \
mongodb_backup_generic.sh
```

2. Для проверки созданных резервных копий на виртуальной машине squadus_infra выполните скрипт:

```
/srv/docker/mongodb_backup_management/backup_scripts/ \
mongodb_restore_generic.sh
```

3. После выполнения скрипта будет запущен контейнер ПО PBM, в котором выполните команду:

```
pbm status
```

Пример вывода:

```
Cluster:
========
shard_2:
  - shard_2/shardsvr-squadus.squadus-db-7.example.com:27018 [S]: pbm-agent v2.3.0
OK
  - shard_2/shardsvr-squadus.squadus-db-8.example.com:27018 [S]: pbm-agent v2.3.0
OK
```

```
 - shard_2/shardsvr-squadus.squadus-db-9.example.com:27018 [P]: pbm-agent v2.3.0
OK
shard_1:
  - shard_1/shardsvr-squadus.squadus-db-4.example.com:27018 [S]: pbm-agent v2.3.0
OK
  - shard_1/shardsvr-squadus.squadus-db-5.example.com:27018 [S]: pbm-agent v2.3.0
OK
  - shard_1/shardsvr-squadus.squadus-db-6.example.com:27018 [P]: pbm-agent v2.3.0
OK
squadus:
  - squadus/configsvr-squadus.squadus-db-1.example.com:27019 [S]: pbm-agent
v2.3.0 OK
  - squadus/configsvr-squadus.squadus-db-2.example.com:27019 [P]: pbm-agent
v2.3.0 OK
  - squadus/configsvr-squadus.squadus-db-3.example.com:27019 [S]: pbm-agent
v2.3.0 OK

PITR incremental backup:
========================
Status [OFF]

Currently running:
==================
(none)

Backups:
========
S3 us-east-1 s3://http://squadus-infra-1.example.com:9000/mongodbbackup/backups
  Snapshots:
    2023-10-22T22:00:02Z 14.23MB <logical> [restore_to_time: 2023-10-
22T22:00:09Z]
    2023-10-22T18:36:51Z 14.20MB <logical> [restore_to_time: 2023-10-
22T18:36:58Z]
    2023-10-21T22:00:02Z 14.55MB <logical> [restore_to_time: 2023-10-
21T22:00:11Z]
    2023-10-20T22:00:02Z 14.55MB <logical> [restore_to_time: 2023-10-
20T22:00:11Z]
```

В списке, выведенном командой, можно увидеть дату выполнения резервного копирования и размер созданного архива.

2.7.1.3 Восстановление

Для восстановления резервной копии следует запустить скрипт:

```
/srv/docker/mongodb_backup_management/backup_scripts/ \
mongodb_restore_generic.sh
```

Для восстановления резервной копии на указанную дату необходимо выполнить команду:

```
pbm restore 2023-10-22T22:00:02Z
```

2.7.1.4 Возможные проблемы

Восстановление резервной копии на чистую базу MongoDB FCV может отличаться. Для изменения версии FCV необходимо подключиться к PRIMARY ноде MongoDB и выполнить команду:

```
db.adminCommand(
   {
      setFeatureCompatibilityVersion: "4.2",
      confirm: true
   }
)
```

После выполнения команды следует продолжить восстановление резервной копии.

2.7.2 Резервное копирование MinIO

Резервное копирование объектного хранилища MinIO выполняется аналогично MongoDB с одним отличием — резервной копией объектного хранилища MinIO является мгновенный снимок состояния без удаления ранее загруженных файлов. Резервное копирование MinIO настраивается ролью `nct.minio.minio_backup` и запускается ежедневно в 2 часа 30 минут. Для изменения расписания запуска необходимо добавить в файл `group_vars/squadus/main.yml` переменные из таблицы 19, и указать час и минуту на запуск.

Таблица 19 — Настройка параметров расписания резервного копирования MinIO

Название переменной	Описание	Тип	Значение по умолчанию
minio_backup_cron_job_hour	Устанавливает час	Int	2
minio_backup_cron_job_minute	Устанавливает минуту	Int	30
minio_backup_dir	Устанавливает директорию для сохранения резервной копии	Str	"/srv/backups/minio{{ ('/' + minio_backup_name) if minio_backup_name }}"
minio_backup_name	Устанавливает название архивной копии	Str	"minio_backup"

Пример корректно настроенных параметров:

```
minio_backup_cron_job_hour: 2
minio_backup_cron_job_minute: 30
```

2.7.3 Применение внесенных изменений в резервное копирование

Для применения изменений на стенде необходимо выполнить команду в рабочей директории `install_squadus`:

```
ansible-playbook playbooks/squadus/backup.yml --diff
```

2.8 Установка в составе других продуктов «МойОфис»

Установка в составе других продуктов «МойОфис» не выполняется.

3 ОБНОВЛЕНИЕ С ПРЕДЫДУЩИХ ВЕРСИЙ

Перед обновлением на всех серверах с ролью `squadus_mail` необходимо выполнить команду:

```
docker rm -fv postfix_relay; rm -rf /srv/docker/postfix_relay/
```

Данный дистрибутив предназначен для чистой установки.

Переход с одной версии на другую осуществляется аналогично установке новой версии. Порядок установки описан в разделе «Первичная установка».

3.1 Обновление системы управления базами данных MongoDB

Перед запуском обновления системы необходимо выполнить следующие действия:

1. Убедиться, что для всех нод группы `squadus_db` в DNS созданы CNAME-записи типа `*.squadus-db-[N].example.net` со значением `squadus-db-[N].example.net`, где `[N]` — соответствующий порядковый номер ноды.
2. Остановить все экземпляры запущенного сервиса `mongodb`, работающие в системе, на каждой из нод группы `squadus_db` с помощью команды:

```
docker stop mongodb && docker rm mongodb
```

3. Переместить содержимое директории `/srv/docker/mongodb/` в `/srv/docker/mongodb_squadus/` на каждой из нод группы `squadus_db` с помощью команды:

```
mv /srv/docker/mongodb/ /srv/docker/mongodb_squadus/
```

4. После этого запустить обновление ПО в обычном режиме.

3.2 Обновление сервиса поиска Squash до версии 1.10

Для проверки версии Squash необходимо на любом сервере группы ролей `squadus_search` выполнить следующую команду:

```
docker ps | grep sqush
```

Перед запуском обновления системы необходимо выполнить следующие действия:

1. Остановить все экземпляры запущенного сервиса `squash_search`, работающие в системе, на каждой из нод группы `squadus_search` с помощью команды:

```
docker stop squash_search
```

2. Удалить директорию на каждой из нод группы `squadus_search`

```
rm -fr /srv/docker/squash_search/data/
```

4 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ И РЕКОМЕНДАЦИИ ПО УСТАНОВКЕ

4.1 Настройка стенда ПО

Для настройки SAML необходимо подключение к контроллеру домена Active Directory Federation Service (далее — ADFS). Для этого следует перейти в доступный через поиск раздел администрирования SAML и заполнить поля (см. Рисунок 4).

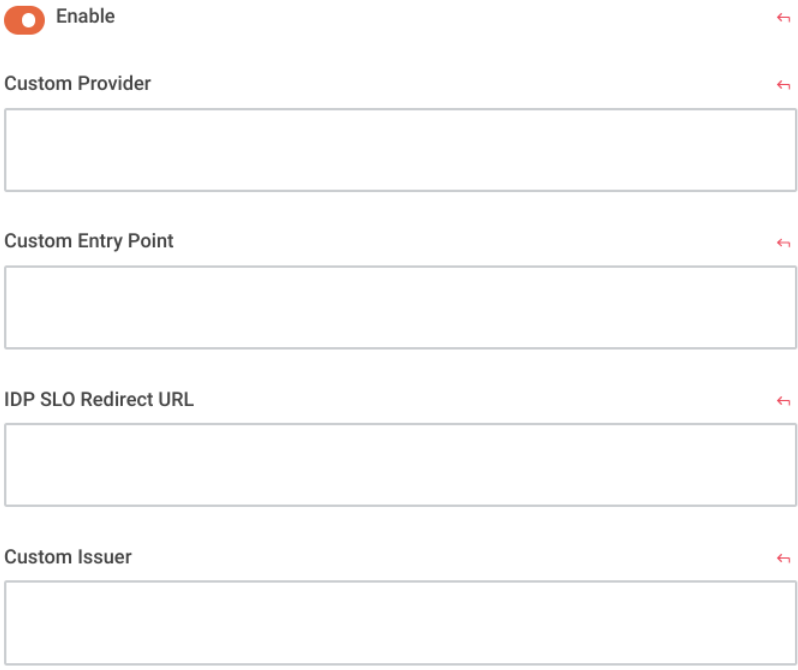


Рисунок 4 — Заполнение полей в SAML разметке

Поля, представленные на рисунке:

- Custom Provider — example-adfs;
- Custom Entry Point — https://mdcad-dc-01.ad.example.ru/adfs/ls/. Адрес контроллера ADFS;
- IDP Slow Redirect URL — https://mdcad-dc-01.ad.example.ru/adfs/ls/. Адрес контроллера ADFS;
- Custom Issuer — https://im.example.ru/_saml/metadata/example-adfs. Ссылка для сбора метаданных стенда. Часть example-adfs заполняется самостоятельно, как и в первом пункте Custom Provider (пункты должны совпадать);
- Private Key Component — приватный ключ. Ключ передается администратором ADFS в формате Base64 (PEM).

4.1.1 Добавление стенда на стороне ADFS

Чтобы добавить стенд на стороне ADFS, необходимо выполнить следующие действия:

1. Открыть панель администрирования ADFS (см. Рисунок 5).

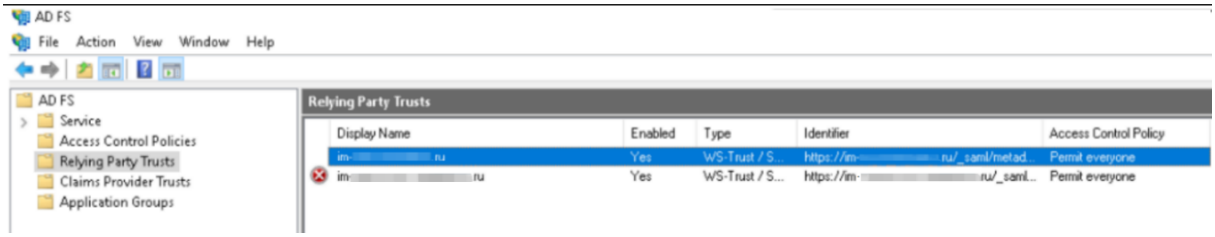


Рисунок 5 — Панель администрирования ADFS

2. В разделе **Relying Party Trust** в правом углу выбрать **Add Relying Party Trust** (см. Рисунок 6).

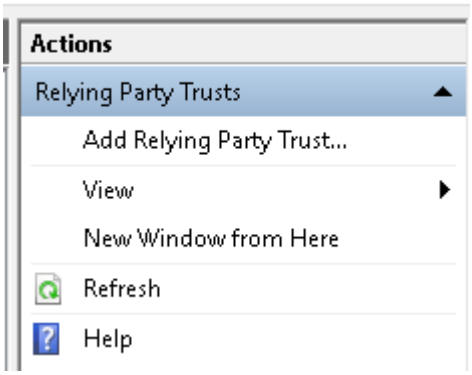


Рисунок 6 — Выбор действия Add Relying Party Trust

3. В первом пункте выбрать **Claims aware > Start**.

4. Далее добавить ссылку, сформированную в поле **Custom Issuer** (см. Рисунок 7).

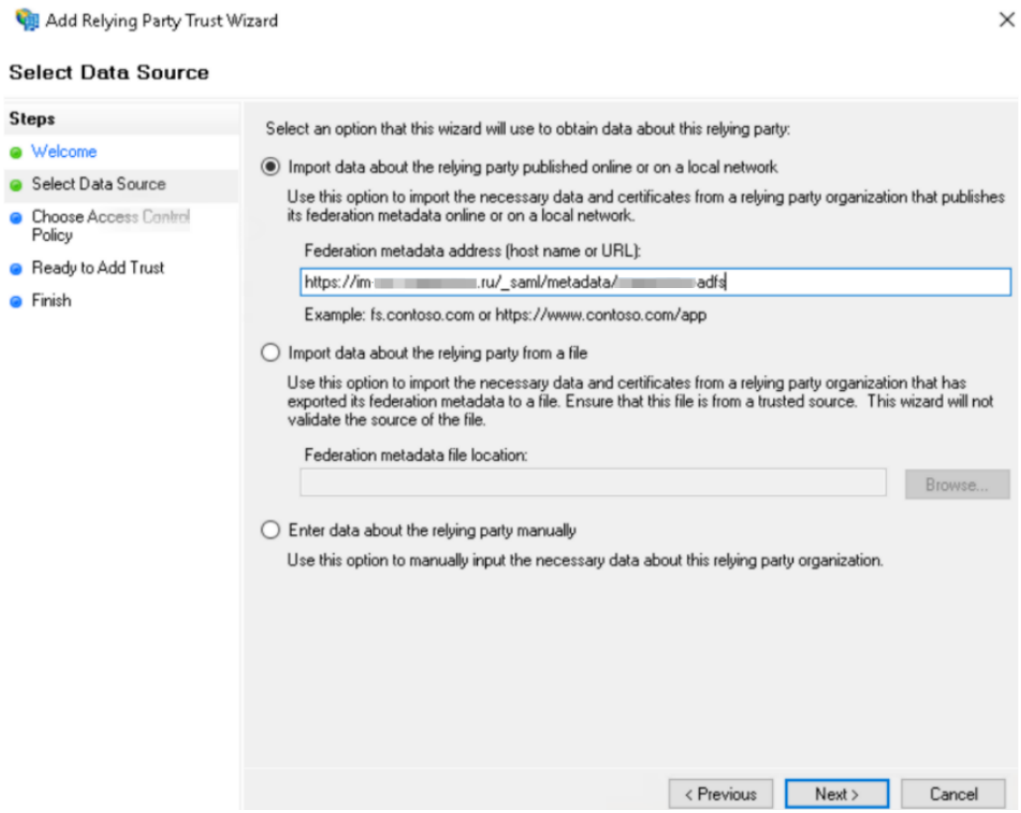


Рисунок 7 — Добавление ссылки

5. В последующих окнах необходимо нажать на кнопку **Next**.

4.1.2 Известные ограничения

Если у пользователя нет полей из Mapping переменных (например, почты) необходимо выполнить следующие действия:

– создать правила для Relying Party Trust для ранее созданного стенда;

– правой кнопкой мыши нажать на созданный ранее стенд **Edit Claim Issuance Policy** (см. Рисунок 8).

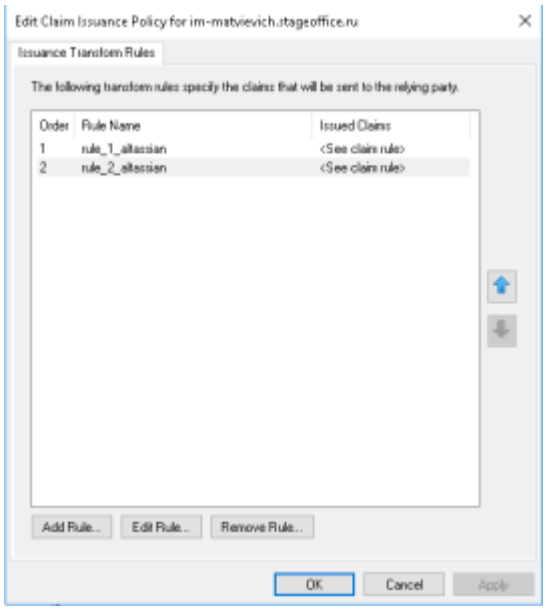


Рисунок 8 — Выбор ранее созданного стенда Edit Claim Issuance Policy

Правило 1:

```
      c:[Type
== "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
Issuer == "AD AUTHORITY"]
       => issue(store = "Active Directory", types =
("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn", "http://schemas.xml
soap.org/ws/2005/05/identity/claims/emailaddress", "http://schemas.xmlsoap.org/ws
/2005/05/identity/claims/givenname", "http://schemas.xmlsoap.org/ws/2005/05/ident
ity/claims/surname"), query = ";objectSID,mail,givenName,sn;{0}", param =
c.Value);
```

Правило 2:

```
      c:[Type
== "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"]
       => issue(Type
= "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier", Issuer
= c.Issuer, OriginalIssuer = c.OriginalIssuer, Value = c.Value, ValueType =
c.ValueType,
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/format
"] = "urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress");
```

Если добавить Relying Party Trust не получается из-за отсутствия поддержки TLS выше версии 1.0 на стороне ADFS, то следует рассмотреть рекомендации по внесению изменений от Microsoft, приведенные в соответствующих статьях.

Чтобы добавить фото пользователя из Active Directory, необходимо выполнить команду:

```
      c:[Type
== "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
Issuer == "AD AUTHORITY"]
       => issue(store = "Active Directory", types = ("thumbnailPhoto"), query
= ";thumbnailPhoto;{0}", param = c.Value);
```

4.2 Настройка вебинаров

4.2.1 Включение вебинаров

По умолчанию данная возможность отключена. Для включения необходимо выполнить следующие действия:

1. Задать переменной `jitsi_webinar_enabled` значение `true` в файле `group_vars/squadus/main.yml`

Пример заполнения:

```
      jitsi_webinar_enabled: true
```

2. Применить конфигурацию с сервера оператора.

```
      ansible-playbook playbooks/squadus.yml --tags nginx,jitsi  --diff
```

3. В разделе **Администрирование** -> **Видеоконференции** -> **Флаги возможностей** включить **Создание видеоконференции в режиме вебинара**.

	Создавать вебинары могут только пользователи с правом **Доступ к созданию вебинаров**.

4.2.2 Проверка работоспособности вебинаров

Для проверки работоспособности необходимо выполнить следующие действия:

1. Запустить ПО.
2. Нажать на пиктограмму  и выбрать в открывшемся меню пункт **Создать вебинар** (см. Рисунок 9).

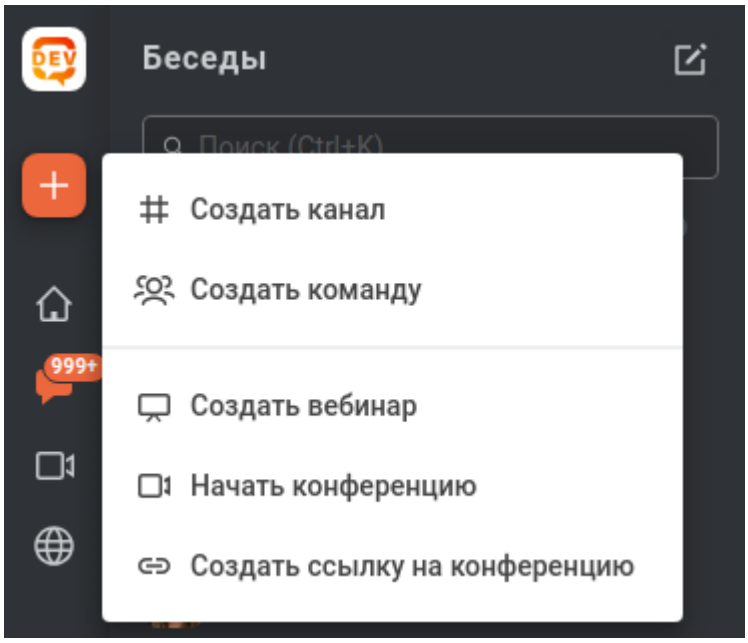


Рисунок 9 — Меню **Создание**

3. В окне создания вебинара (см. Рисунок 10) необходимо указать **Тему** и присоединиться по указанной ссылке.

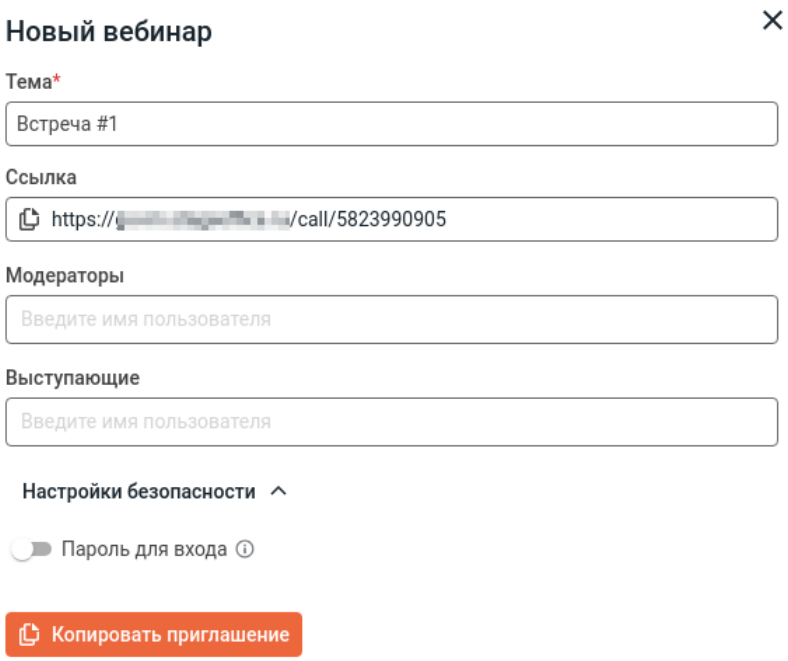


Рисунок 10 — Окно создания нового вебинара

4.3 Интеграция на примере автоматической телефонной станции Asterisk

Этот пример демонстрирует возможность подключения цифровой автоматической телефонной станции к компоненту ПО — SIP-шлюзу. Это позволяет участвовать в конференциях с помощью телефона, поддерживающего тоновый набор номера.

Подключение к конференции осуществляется звонком с телефона на заранее настроенный номер на цифровой АТС (в этом примере используется номер «100»).

Чтобы попасть в конференцию, клиенту необходимо ввести ПИН (идентификатор) конференции, используя систему голосового меню IVR.

Приведенные конфигурационные файлы АТС Asterisk являются примерами и не предназначены для использования в production-среде, а также ориентированы на версию Asterisk 18.

На рисунке приведена схема взаимодействия компонентов ПО и АТС Asterisk (см. Рисунок 11).

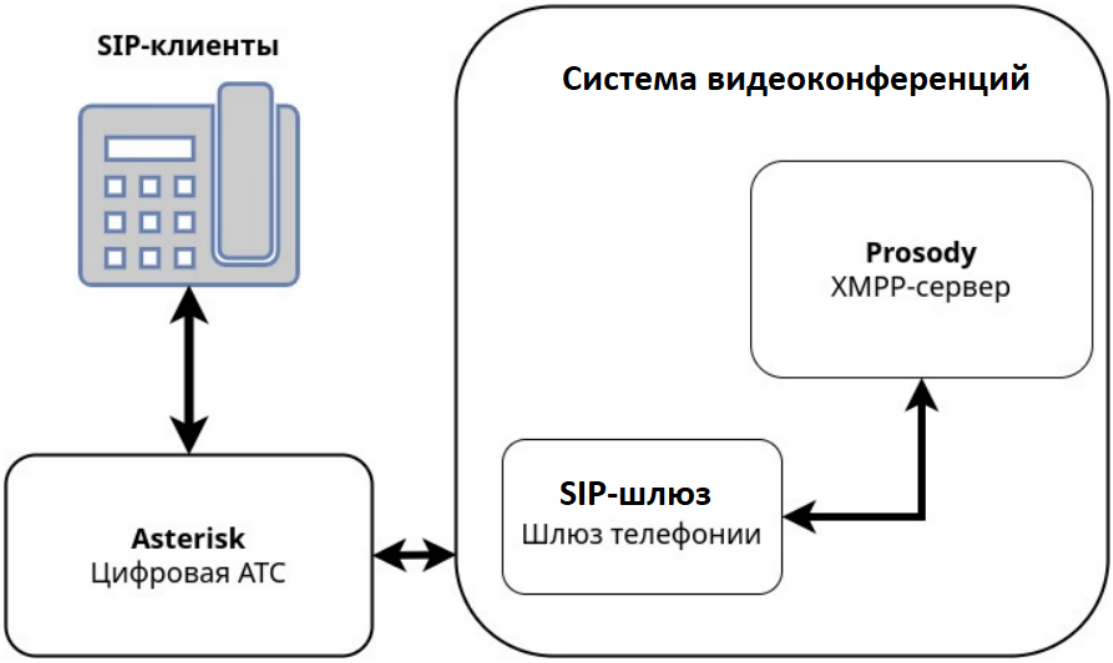


Рисунок 11 — Схема взаимодействия компонентов ПО и АТС Asterisk

4.3.1 Процедура настройки взаимодействия компонентов с Asterisk

Для настройки взаимодействия необходимо выполнить следующие действия:

1. Включить интеграцию SIP-шлюз для Jitsi, а также задать другие необходимые переменные в файле `group_vars/squadus_setup/extra_vars.yml`:

```
      jitsi_sip_enabled: true
      jigasi_sip_server: "10.0.0.1"
      jigasi_sip_uri: "jigasi@{{ jigasi_sip_server }}"
      jigasi_auth_password: "SECURE_PASSWORD_GOES_HERE"
      jigasi_sip_password: "ANOTHER_SECURE_PASSWORD_GOES_HERE"
```

2. Настроить пользователя Asterisk с SIP номером 1001 в конфигурационном файле `pjsip.conf`:

```
      [transport-jigasi]
      type = transport
      protocol = udp
      bind = 0.0.0.0:5160

      ; внешний адрес хоста
      external_media_address=10.0.0.1
      external_signaling_address=10.0.0.1

      [jigasi_auth]
      type = auth
      auth_type = userpass

      ; Пароль в незашифрованном виде. Такой же, как в переменной
jigasi_auth_password в файле extra_vars.yml
      password = SECURE_PASSWORD_GOES_HERE
      username = jigasi

      [jigasi]
      type = aor
      max_contacts = 1
      remove_existing = yes

      [jigasi]
      type = endpoint
      transport = transport-jigasi
      ;context = jigasi-in

      ; public
      context = public
      direct_media = no
      disallow = all
      allow = speex
      auth = jigasi_auth
      aors = jigasi
      rtp_symmetric=yes
      force_rport=yes

      rewrite_contact=yes

      [1001]
      type = aor
      max_contacts = 1
```

```
[1001]
type = auth
username = 1001
password = 1234
[1001]
type = endpoint
transport = transport-jigasi

; public
context = public
auth = 1001
outbound_auth = 1001
aors = 1001
disallow = all
allow = g722
allow = speex
allow = alaw
```

3. Задать диапазон портов для RTP протокола в конфигурационном файле `rtp.conf`:

```
rtpstart=10050
rtpend=10100
```

4. Настроить интерактивное голосовое меню IVR в конфигурационном файле extensions.conf:

```
[public]
exten => 1001,1,Dial(PJSIP/1001,10)
exten => jigasi,1,Dial(PJSIP/jigasi,10)

exten => 100,1,Answer()
same => n,Set(CHANNEL(language)=ru)
same => n,Playback(greeting)
same => n,Set(Attempts=0)
same => n(getmeeting),Set(Attempts=${MATH(${Attempts}+1,i)})
same => n,Verbose(Attempts=: ${Attempts})
same => n,ExecIf($["${Attempts}" = "4"]?Gosub(Attempts-Error,s,1))
same => n,ExecIf($["${Attempts}" != "1"]?Playback(tryagain))
same => n,Read(PIN,beep,12)
same => n,Verbose(Result is: ${PIN})
same => n,GotoIf($[ "${PIN}" == "" ]?getmeeting)
same => n,Read(confpassword,enterpass&beep,6,,,10)
same => n,Verbose(ResultPASS is: ${confpassword})
same => n,AGI(conferenceMapper.sh,${PIN})
same => n,Verbose(Result is: ${ROOM})
same => n,GotoIf($[ "${ROOM}" == "false" ]?invalidnum:joinmeeting)
same => n(invalidnum),Playback(simplewrongconfid)
same => n,Goto(getmeeting)
same => n(joinmeeting),Playback(thanks)
same => n,Playback(enterconf)
same => n,Verbose(Begin Result is:  ${ROOM} -Pass: ${confpassword})
same => n,Dial(PJSIP/jigasi,,b(sub-
headers^caller_handler^1(${ROOM},${confpassword})))
same => n,Verbose(0, Contacting Jitsi... Status is ${DIALSTATUS} );
same => n,Dial(PJSIP/jigasi,,b(sub-
headers^caller_handler^1(${ROOM},${confpassword})))
same => n,Verbose(0, Contacting Jitsi... Status is ${DIALSTATUS} );
```

```
      [easybell-in]
      exten => _X.,1,Dial(PJSIP/jigasi,,b(handler^addheader^2))
      exten => _X.,n,Playback(hello)
      exten => _X.,n,Playback(conf-getconfno)
      exten => _X.,n(getmeeting),Read(PIN,beep,12)
      exten => _X.,n,Verbose(Result is: ${PIN})
      exten => _X.,n,AGI(conferenceMapper.sh,${PIN})
      exten => _X.,n,Verbose(Result is: ${ROOM})
      exten => _X.,n,GotoIf($[ "${ROOM}" == "false" ]?invalidnum:joinmeeting)
      exten => _X.,n(invalidnum),Playback(conf-invalid)
      exten => _X.,n,Goto(getmeeting)
      exten => _X.,n(joinmeeting),Playback(conf-placeintoconf)
      exten => _X.,n,Dial(PJSIP/jigasi,,b(sub-
headers^caller_handler^1(${ROOM},${PIN})))

      [sub-headers]
      exten => caller_handler,1,NoOp(Set Header Jitsi-Conference-Room: ${ARG1} -
Pass: ${ARG2})
      same => n,Verbose(Result is:  ${ARG1} -Pass: ${ARG2} )
      same => n,Set(PJSIP_HEADER(add,Jitsi-Conference-Room)=${ARG1})
      same => n,Set(PJSIP_HEADER(add,X-Room-Name)=${ARG1})
      same => n,Set(PJSIP_HEADER(add,Jitsi-Conference-Room-Pass)=${ARG2})
      same => n,Verbose(RETURN!!!!)
      same => n,Return(${DIALSTATUS})
```

5. Создать AGI-скрипт, а также выставить необходимые права и сделать его исполняемым:

```
      cat << 'EOF' > /var/lib/asterisk/agi-bin/conferenceMapper.sh
      #!/bin/sh

      if command -v curl 1>/dev/null; then
          HTTP_CLIENT="curl -s"
      elif command -v wget 1>/dev/null; then
          HTTP_CLIENT="wget -q -O -"
      else
          echo >&2 "ERROR! HTTP client is absent! Please Install curl or wget."
          exit 1
      fi

      jitsi_room=$($HTTP_CLIENT
http://CHANGE_IT_TO_CONFERENCE_MAPPER_HOST:8001/conferenceMapper?id=$1 | sed -r
's#^.*conference":"(\d+)@.*#\1#g')

      echo "SET VARIABLE ROOM \"${jitsi_room}\" "
      EOF

      chown asterisk:asterisk /var/lib/asterisk/agi-bin/conferenceMapper.sh

      chmod 750 /var/lib/asterisk/agi-bin/conferenceMapper.sh
```

6. Настроить конфигурацию Nginx. Для этого в файле `group_vars/squadus_ha/main.yml` добавить в структуру данных `nginx_vhosts.meet` такие пары ключ-значение:

```
nginx_vhosts:
  "meet":
    dialin_conference_mapper_backends: "{{ [groups['squadus_infra'][0]] }}"
    dialin_enabled: true
    dialin_phone_numbers:
      message: "Phone numbers available."
      numbers:
        RU:
          - "100"
      numbersEnabled: true
```

7. Развернуть сервис `conferencemapper` на тех же хостах, что указаны в переменной `nginx_vhosts.meet.dialin_conference_mapper_backends` в предыдущем пункте. При этом ожидается, что сервис будет доступен по TCP-порту 8001. В случае необходимости его можно переопределить переменной `conference_mapper_port`.

8. Применить конфигурацию:

```
[root@squadus_infra~]# ansible-playbook playbooks/squadus.yml \
 --diff -t  nginx,jitsi
```

4.3.2 Проверка работоспособности

Для проверки работоспособности необходимо выполнить следующие действия:

1. Подключиться SIP-клиентом к хосту, на котором был развернут АТС Asterisk. Для аутентификации использовать логин и пароль, сконфигурированный ранее в файле `pjsip.conf`.

2. Создать конференцию Jitsi и найти идентификатор, он же ПИН (см. пункт 3, Рисунок 12).

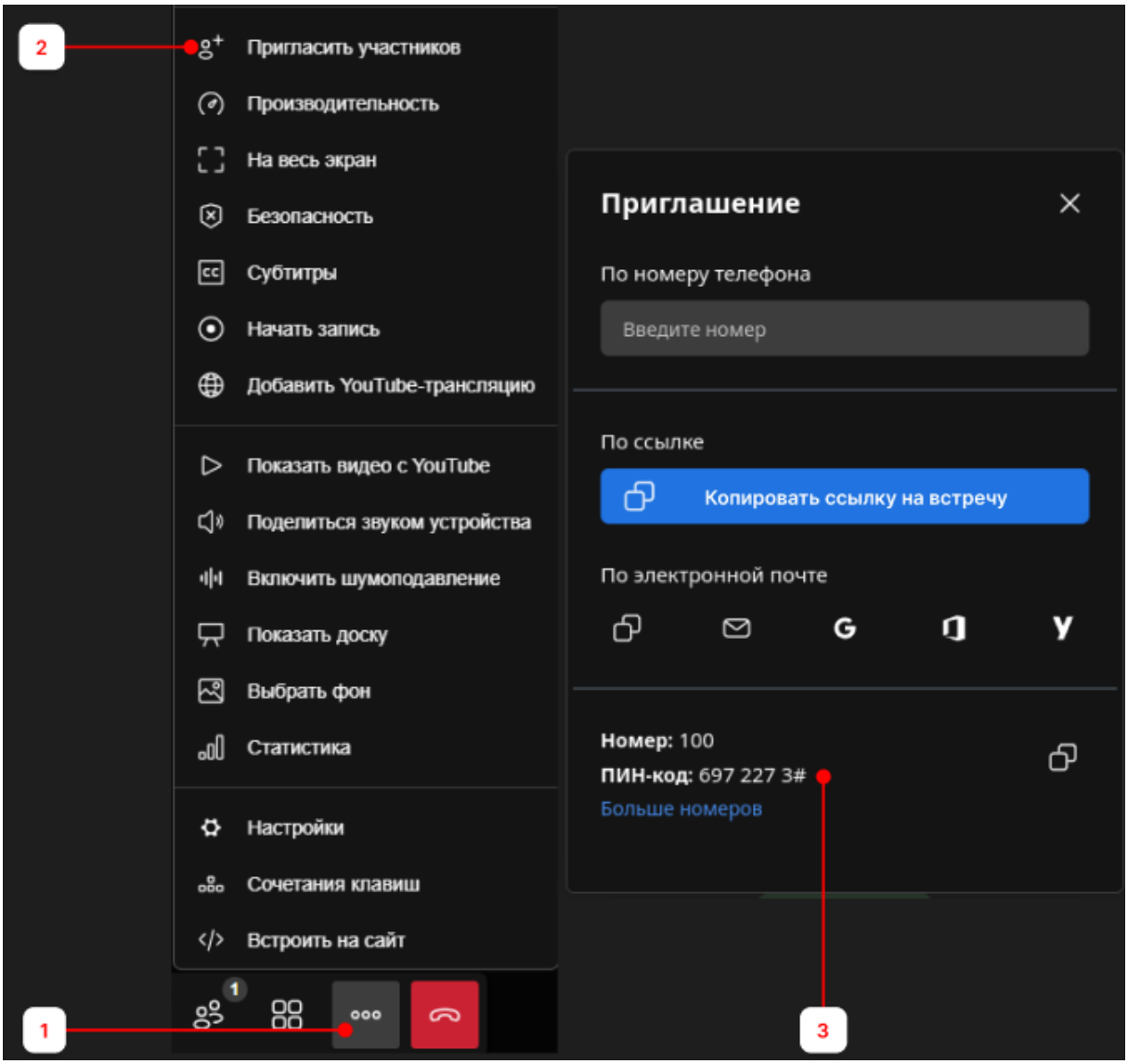


Рисунок 12 — Поиск идентификатора

3. Позвонить на сконфигурированный ранее в файле `extensions.conf` номер 100 SIP клиента.

4. Используя инструкции голосового меню, ввести идентификатор конференции. Так как конференция в данном случае не защищена паролем, в голосовом меню на этапе ввода пароля необходимо нажать решетку.

В результате к конференции должен присоединиться участник с именем 1001 (см. Рисунок 13).

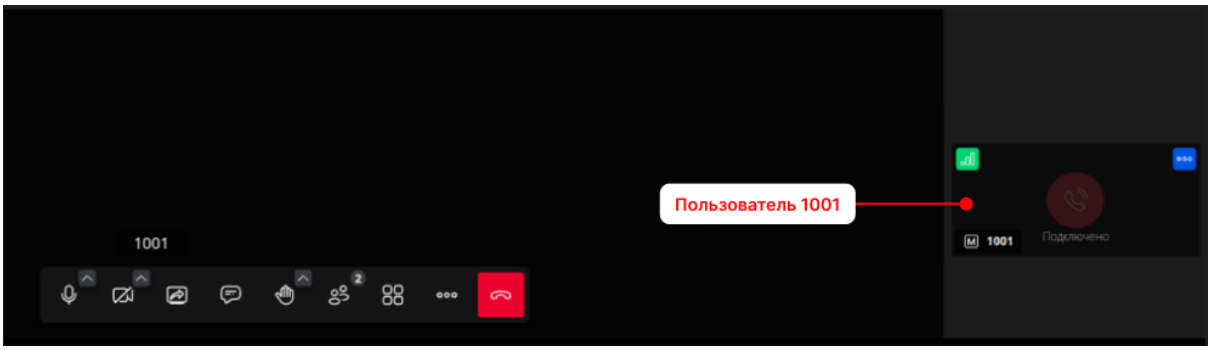


Рисунок 13 — Участник с именем 1001

4.4 Настройка виртуальной доски для совместного использования

В ПО реализована возможность использования виртуальной доски во время видеоконференций. Совместный доступ позволяет каждому участнику рисовать на виртуальной доске — процесс виден всем участникам в реальном времени. Результат можно сохранить в формате PNG или SVG непосредственно из конференции.

В данном релизе виртуальная доска не видна пользователям мобильных устройств.

4.4.1 Включение виртуальной доски

По умолчанию данная возможность отключена. Для включения необходимо выполнить следующие действия:

Задать переменной `excalidraw_whiteboard_enabled` значение `true` в файле `group_vars/squadus/main.yml`. Подробная информация приведена в разделе «Настройка дополнительных параметров установки».

```
excalidraw_whiteboard_enabled: true
```

Применить конфигурацию. Данное действие должно выполняться с подготовленного места оператора, которое должно соответствовать требованиям, описанным в разделе «Установка подсистемы управления конфигурациями (Ansible)».

```
ansible-playbook playbooks/squadus.yml --tags whiteboard --diff
```

После применения конфигурации в видеоконференциях появится возможность включения и совместного использования доски (см. Рисунок 14).

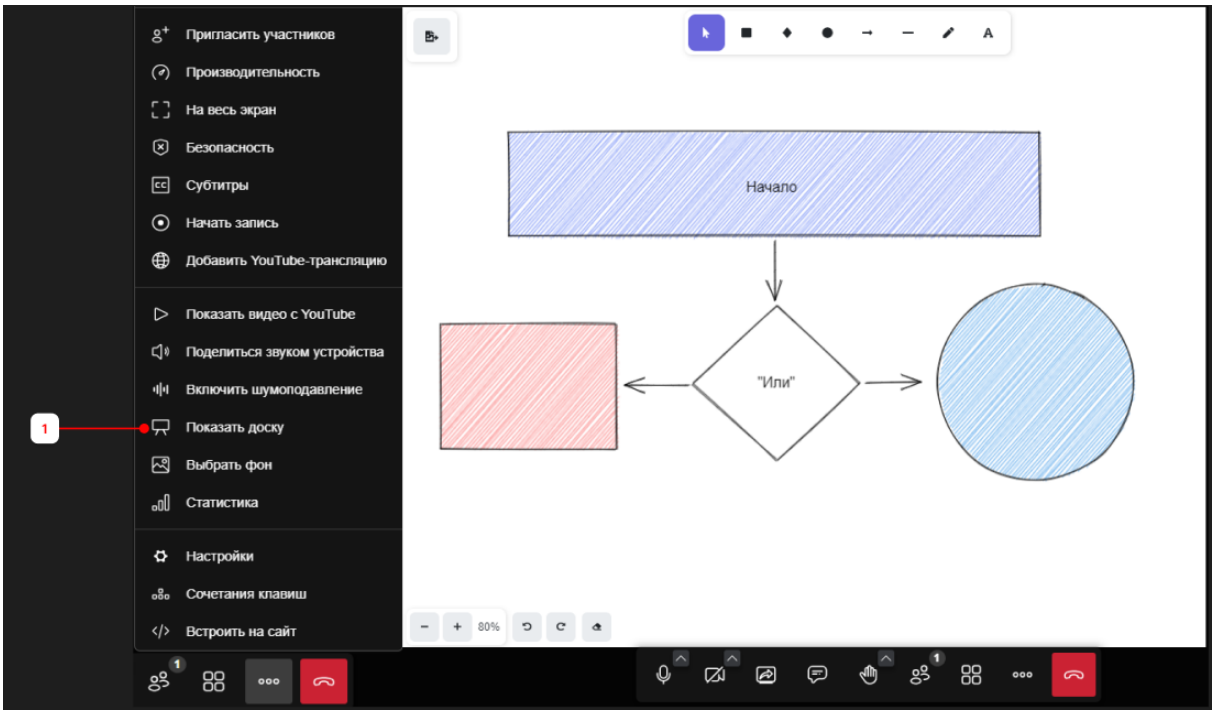


Рисунок 14 — Функция отображения доски для совместного использования

4.5 Настройка транскрибации речи (субтитры)

В ПО реализована возможность транскрибации аудиопотока видеоконференций в реальном времени. Такой режим позволяет распознавать речь участников и выводить результат в текстовом формате в окне конференции (см. Рисунок 15).

Включить данную функцию может любой участник конференции. Результат транскрибации виден только тому участнику, который включил данную функцию.

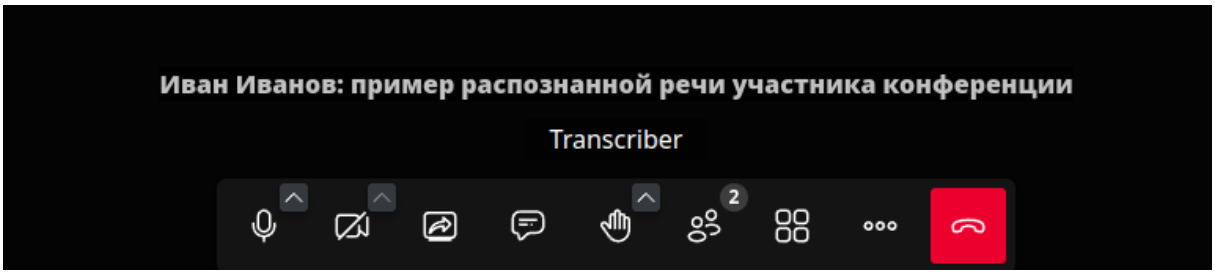


Рисунок 15 — Функция транскрибации

4.5.1 Системные требования

Модели для транскрибации являются высокоточными и могут использовать значительные ресурсы для функционирования. Рекомендуется использовать следующие параметры: не менее 4 vCPU и 16 Гбайт RAM для одной модели.

4.5.2 Включение транскрибации речи

По умолчанию транскрибация речи отключена. В текущей версии поддерживаются русский и английский языки. Но одновременно может использоваться только один из них.

Для установки модели для русского языка и включения транскрибации речи необходимо:

1. Предварительно загрузить в Docker-registry контейнеры языковых моделей из архивов. Архивы можно найти в поставке дистрибутива, в файлах-архивах `vosk_[LANG]_[X_X].tar`, где `[LANG]` — это идентификатор языка, а `[X_X]` — это версия ПО. Пример общего алгоритма загрузки приведен ниже:

```
# загрузить образ файла-архива
docker load -i vosk_ru_1_2.tar

# поставить тег
# предварительно заменить "docker_registry_fqdn" на актуальный fqdn хоста с Docker-registry, по умолчанию это "{{ groups['squadus_infra'][0] }}:5000"
docker tag vosk-ru:1.2 docker_registry_fqdn/vosk-ru:1.2

# выгрузить образ в Docker-registry
docker push vosk-ru:1.2
```

2. В файле `inventory/group_vars/squadus/main.yml` задать переменной `vosk_enabled` значение `true`, а также определить переменную `vosk_hosts`.

```
vosk_enabled: true
vosk_hosts:
  vosk.example.net
    models:
      - language: "ru"
        listen_tcp_port: 2700
```

3. Добавить в файл inventory в раздел группы `squadus_meet_vosk` хост, на который планируется устанавливать модели VOSK.

```
squadus_meet_vosk:
hosts:
      vosk.example.net:
```

4. Применить конфигурацию:

```
ansible-playbook playbooks/squadus.yml --tags jigasi,vosk,jitsiweb --diff
```

Для установки модели для английского языка, необходимо определить переменную `jitsiweb_transcription_preferred_language` со значением "en-US", а также задать переменной `vosk_hosts['vosk.example.net']['models'][0].language` значение `"en"`:

```
jitsiweb_transcription_preferred_language: "en-US"
vosk_enabled: true
vosk_hosts:
  vosk.example.net
    models:
      - language: "en"
          listen_tcp_port: 2700
```

5. Выполнить пункт 3 из раздела «Проверка работоспособности».

Для включения транскрибации необходимо в меню конференции выбрать пункт **Субтитры** (см. Рисунок 16), после чего появится одноименное модальное окно, в котором следует перевести переключатель в состояние **Вкл**.

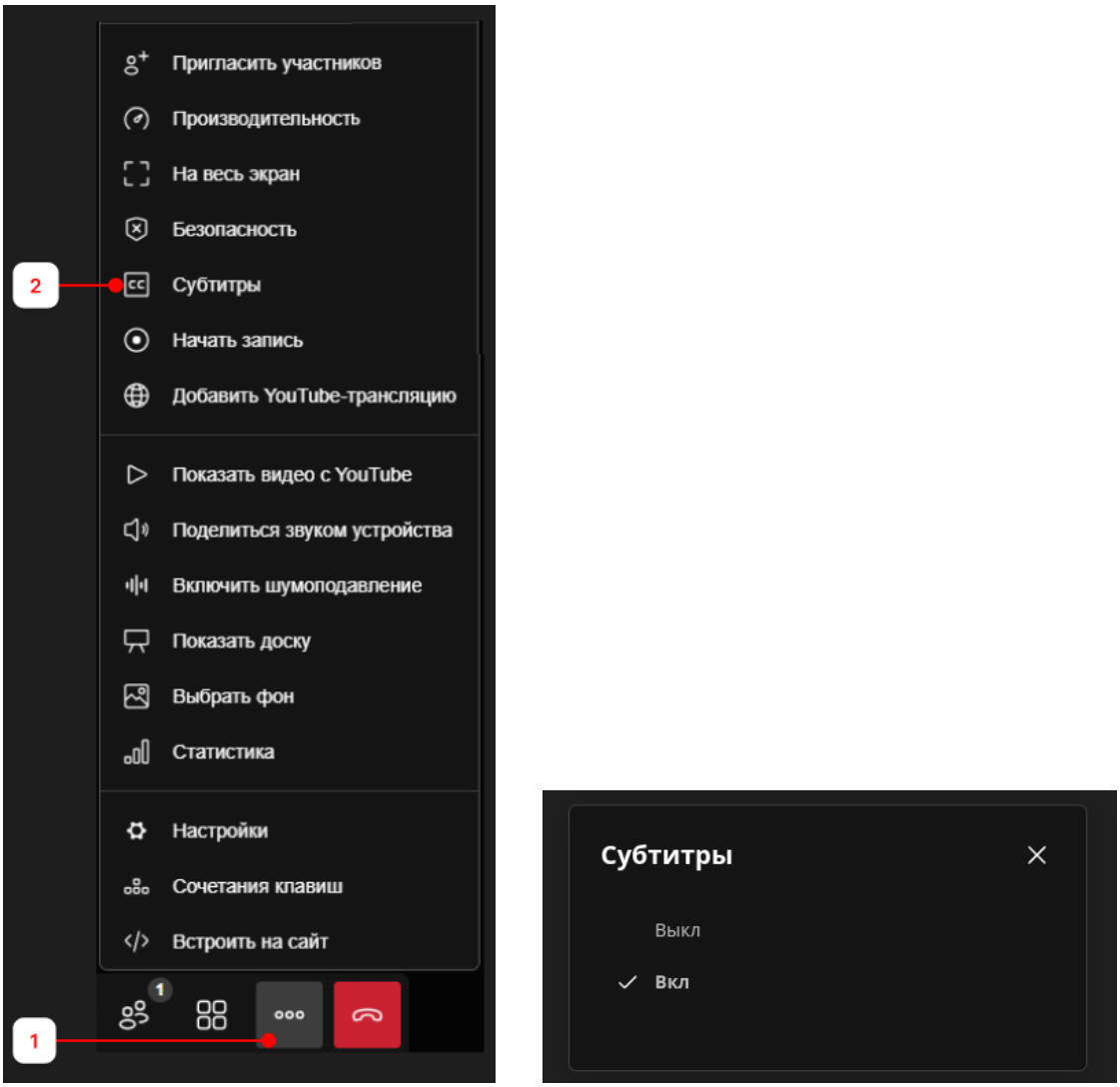


Рисунок 16 — Включение транскрибации

4.6 Интеграция с Jira

В ПО используется сервис `squadus-jiratrigger`, позволяющий сформировать в предварительный просмотр задачи в Jira. Сервис обращается к Jira на основании идентификатора задачи в сообщении пользователя. Таким образом, в канале/треде, в котором поступило сообщение, задача отобразится с номером в виде дополнительного сообщения.

Также сервис отправляет краткую сводку по изменению в проекте. В ранее выпущенных дистрибутивах используется сервис `jiratrigger` для формирования предварительного просмотра задач Jira. В новых версиях дистрибутива ПО данный сервис замещается сервисом `squadus_jiratrigger`.

4.6.1 Настройка Jira для интеграции с ПО

Для интеграции с Jira необходимо выполнить следующие действия:

- создать пользователя в Jira;
- установить пароль созданному пользователю;
- выдать права пользователю на чтение задач проекта.

Для создания пользователя и присвоения прав необходимо использовать учетную запись с правами администратора.

4.6.2 Настройка сервиса squadus_jiratrigger

Для установки сервиса `squadus_jiratrigger` необходимо присвоить переменной `squadus_jiratrigger_enabled` значение `true` в файле `inventory/group_vars/squadus/main.yml`.

Указывать значения переменных следует аналогично настройкам Jira, в таблице 20 приведен пример настройки параметров.

Таблица 20 — Настройка параметров сервиса squadus_jiratrigger

Параметр	Описание	Тип	Значение по умолчанию
squadus_jiratrigger_enabled	Включение сервиса squadus_jiratrigger	Bool	false
squadus_jiratrigger_jira_password	Пароль пользователя для работы с Jira	Str	"5F5H6Fh3XSMcbZLaSbFvxbfXlkDbPCuU2"
squadus_jiratrigger_jira_url	URL сервиса Jira	Str	"https://jira.com"
squadus_jiratrigger_jira_username	Имя пользователя для работы с Jira	Str	"JiraUser"

Пример настроенных параметров:

```
squadus_jiratrigger_jira_password: "5F5H6Fh3XSMcbZLaSbFvxbfXlkDbPCuU2"
squadus_jiratrigger_jira_url: "https://jira.com"
squadus_jiratrigger_jira_username: "JiraUser"
```

Если переменные не были определены заранее в файле `contrib/squadus/group_vars/squadus_setup`, то необходимо запустить playbook Ansible с нижеприведенными параметрами для установки сервиса `squadus_jiratrigger`.

```
ansible-playbook -i hosts.yml playbooks/squadus/squadus_apps.yml -l squadus_apps -t squadus_jiratrigger
```

После выполнения команды, на серверах группы `squadus_apps` будет запущен контейнер с названием `squadus_jiratrigger`.

4.6.3 Настройки сервиса jiratrigger

Настройки сервиса `jiratrigger` отличаются исключительно переменными в Ansible playbook, настройки интеграции сервиса `squadus` с Jira производятся аналогичным образом.

Для установки сервиса необходимо отредактировать значения переменных в файле `inventory/group_vars/squadus/main.yml`. Переменной `squadus_enable_jiratrigger` необходимо присвоить значение `true`. Переменные с указанием `login`, `password`, `url` потребуется добавить в файл. Пример настройки параметров приведен в таблице 21.

Таблица 21 — Настройки сервиса jiratrigger

Параметр	Описание	Тип	Значение по умолчанию
squadus_enable_jiratrigger	Включение сервиса squadus_jiratrigger	Bool	false
jiratrigger_jira_password	Пароль пользователя для работы с Jira	Str	"5F5H6Fh3XSMcbZLaSbFvxbfXlkDbPCuU2"
jiratrigger_jira_url	URL сервиса Jira	Str	"https://jira.com"
jiratrigger_jira_username	Имя пользователя для работы с Jira	Str	"JiraUser"

Пример настроенных параметров:

```
squadus_enable_jiratrigger: true
jiratrigger_jira_password: "5F5H6Fh3XSMcbZLaSbFvxbfXlkDbPCuU2"
jiratrigger_jira_url: "https://jira.com"
jiratrigger_jira_username: "JiraUser"
```

Для установки сервиса необходимо запустить команду:

```
[root@squadus_infra~]# ansible-playbook -i hosts.yml playbooks/squadus/squadus_apps.yml -l squadus_apps -t jiratrigger
```

После выполнения команды на серверах группы `squadus_apps` будет запущен контейнер с названием `jiratrigger`.

4.6.4 Возможные проблемы

Если в системе ранее был установлен сервис `jiratrigger`, playbook Ansible автоматически удалит старый сервис.

При возникновении проблем во время установки нового сервиса необходимо проверить занятость порта 4567/tcp на серверах группы `squadus_apps`. Если старый сервис не был удален автоматически, его необходимо удалить на каждом сервере группы с помощью команды:

```
docker rm -f jiratrigger
```

4.7 Настройка автоматических уведомлений в проекте Jira

Для автоматического уведомления о новых задачах и изменении статуса задачи в проекте Jira необходимо настроить webhook. При создании или изменении статуса задачи, пользователь с ролью `bot` будет отправлять сообщение с кратким описанием в указанный канал.

4.7.1 Настройка входящих webhook

Настройка входящих webhook осуществляется пользователем с правами администратора.

Для настройки входящего webhook необходимо:

1. Перейти в раздел **Администрирование > Интеграции**.
2. Нажать кнопку **Новый**.
3. Перейти на закладку **Incoming**.

В таблице 22 приведены обязательные параметры для заполнения.

Таблица 22 — Настройка параметров

Параметр	Описание	Значение
Включено	Включает webhook	Активация
Имя	Узнаваемое описание webhook	Название проекта Jira
Опубликовать в канале	Название канала для публикации уведомлений	#channel_name
Отправить от имени	Имя пользователя с ролью bot	squadus.bot
Использовать скрипт	Включает скрипт обработки входящих webhook	Активация
Script	Скрипт обработки webhook	Скрипт приведен ниже

Параметр	Описание	Значение
URL-адрес webhook	URL webhook для Jira	Формируется автоматически
Токен	Токен для Jira	Формируется автоматически

Скрипт обработки входящих webhook:

```
/*jshint esnext:true*/
class Script {
process_incoming_request({request}) {
const data = request.content;
try {
let issue = data.issue;
let baseJiraUrl = issue.self.replace(/\/rest\/.*$/, '');
const result = {
content: {
attachments: [
{
author_name: issue.key,
author_icon: issue.fields.priority.iconUrl,
thumb_url: issue.fields.issuetype.iconUrl,
text: `[${issue.fields.summary}](${baseJiraUrl}/browse/${issue.key})`,
},
{
author_name: `Creator: ${data.user.displayName}`,
author_icon: data.user.avatarUrls['24x24'],
},
{
author_name: `Assignee: ${data.issue.fields.assignee ?
data.issue.fields.assignee.displayName : 'не назначен'}`,
author_icon: (data.issue.fields.assignee &&
data.issue.fields.assignee.avatarUrls['24x24']) || undefined,
}
]
}
}
return result;
} catch(e) {
return { content: { text: `${e.message || e} ${JSON.stringify(data)}`}};
}
}
}
```

На рисунках приведены примеры заполнения полей (Рисунок 17, Рисунок 18).

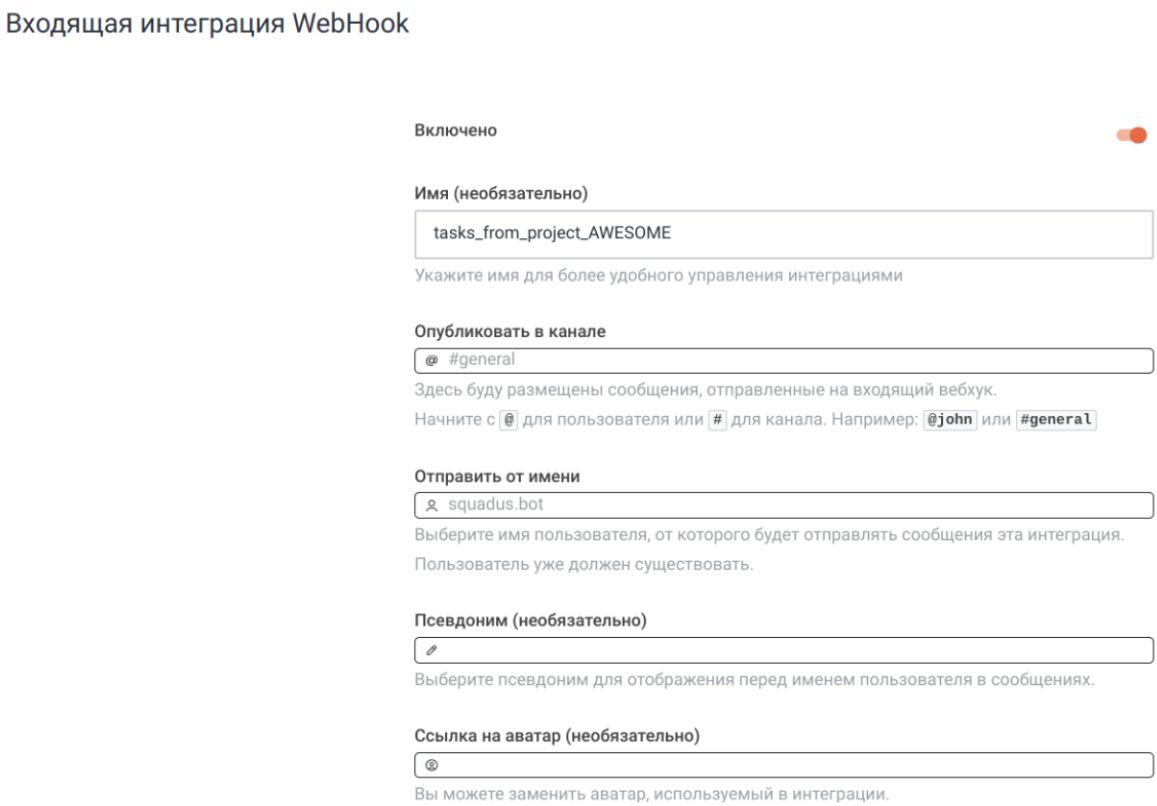


Рисунок 17 — Пример заполненных полей в приложении

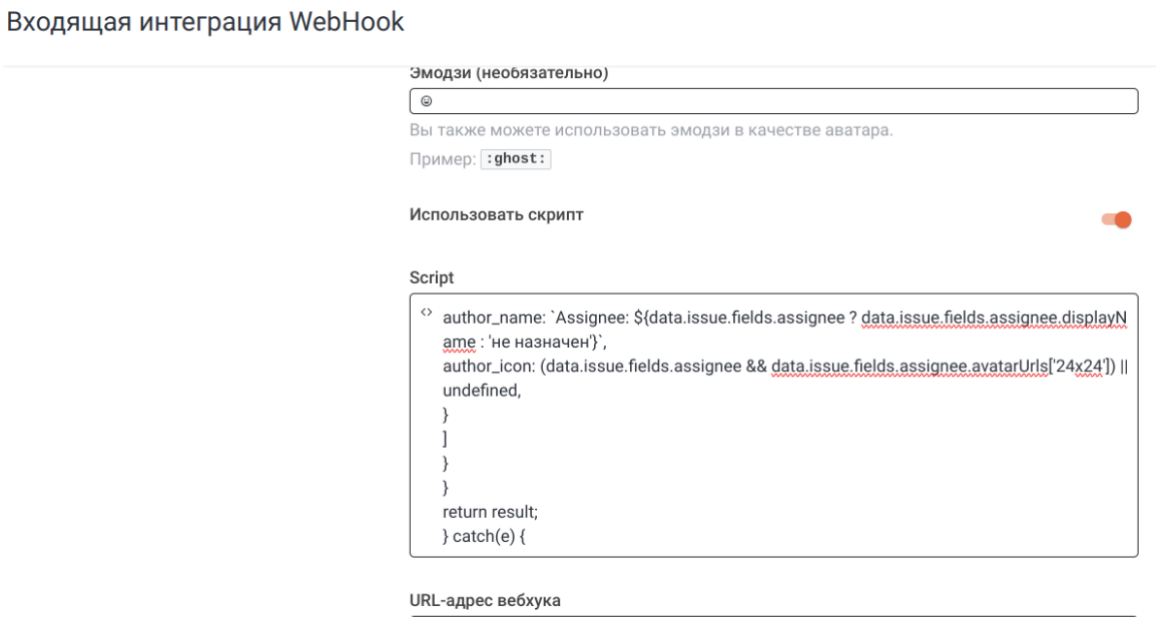


Рисунок 18 — Пример заполненных полей в приложении

После заполнения обязательных полей необходимо выполнить следующие действия:

1. Нажать на кнопку **Сохранить**.
2. Повторно открыть созданный webhook и сохранить значения полей URL-адрес webhook и токен для настройки на стороне Jira.

Данные поля будут заполнены автоматически после сохранения настроек webhook.

4.7.2 Настройка webhook Jira

Для настройки уведомлений о создании или изменении статуса задачи в проекте Jira необходимо создать webhook.

Для применения настроек у пользователя должны быть права администратора Jira. Для этого необходимо перейти в **Администрирование > Система > webhook** и заполнить обязательные параметры из таблицы 23.

Таблица 23 — Настройка параметров

Параметр	Описание	Значение
Имя	Название webhook	Узнаваемое описание
Статус	Включено	Включает webhook
URL	URL webhook	URL-адрес webhook из настроек в ПОприложении
События	Выборка задач из проекта	JQL позволяющий сформировать условия выборки задач из проекта

На рисунках 19 и 20 приведены примеры заполнения полей.

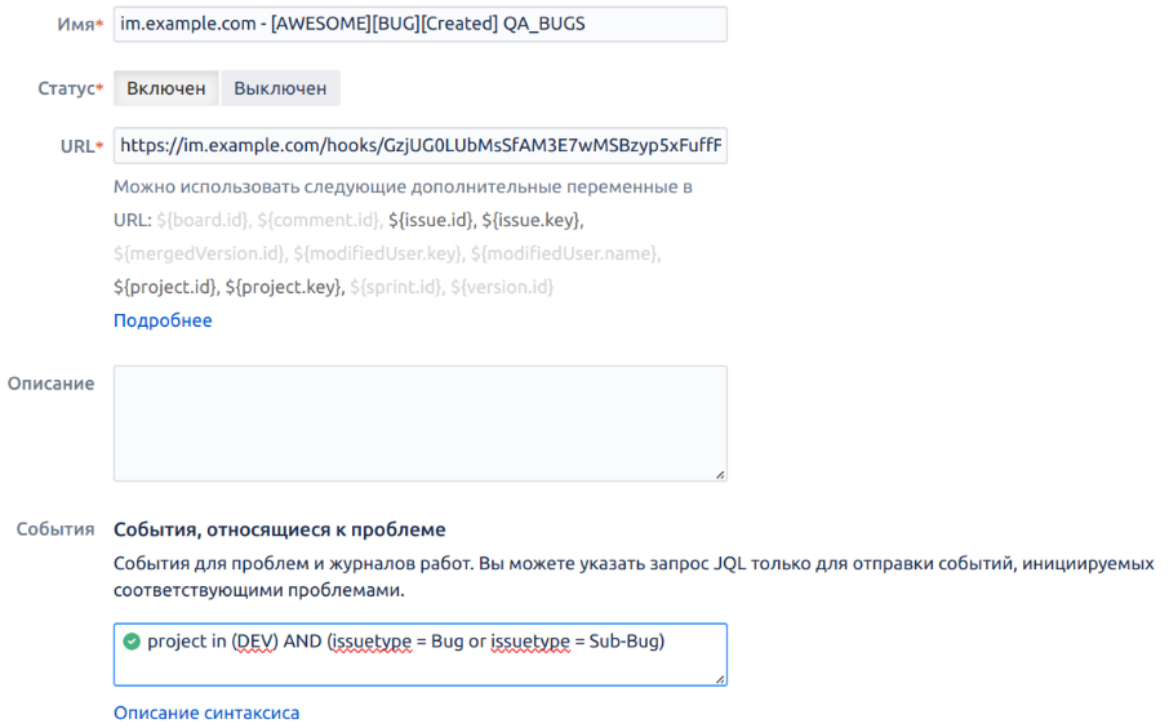


Рисунок 19 — Пример заполненных полей в Jira

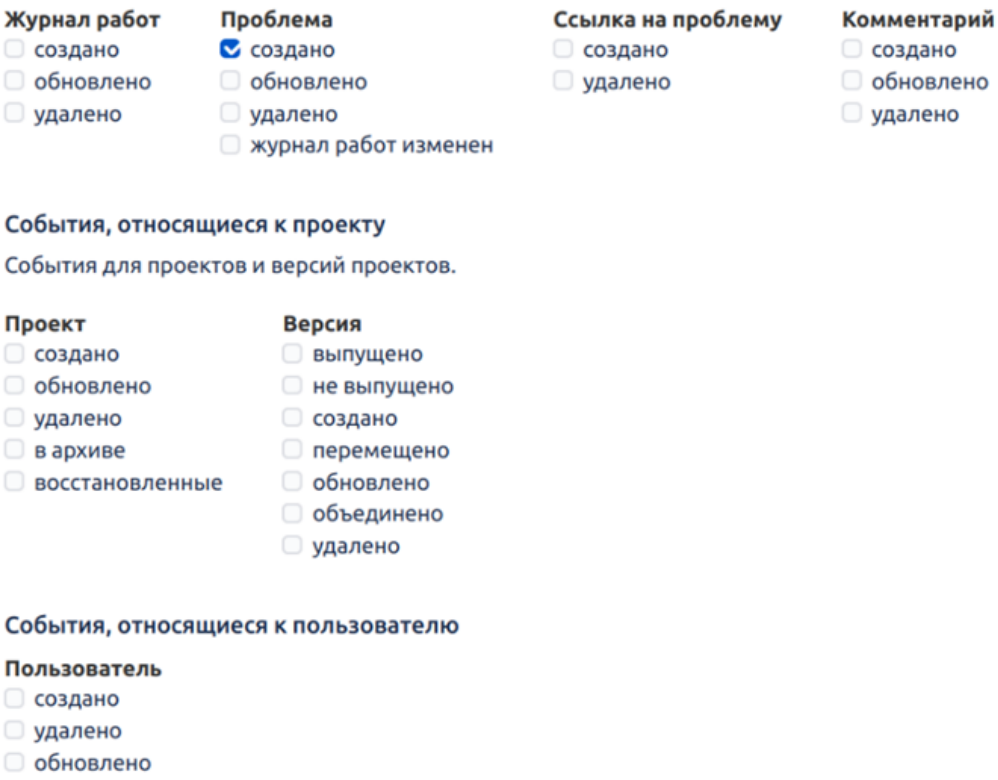


Рисунок 20 — Пример заполненных полей в Jira

После заполнения полей необходимо нажать кнопку **Сохранить**.

	В примере создан webhook для проекта DEV, задач типа BUG и рассылкой уведомлений при создании задачи в проекте. Поле URL заполняется ссылкой автоматически сформированной в приложении в поле URL-адрес webhook.

4.8 Режим федерации

Режим федерации в приложении — это возможность создавать комнаты/каналы между разными серверами приложения и приглашать в комнаты пользователей из разных серверов. Взаимодействие серверов приложения в режиме федерации основано на сервисе Matrix Synapse (который также может называться homeserver), и устанавливается на каждый сервер.

В качестве базы данных Matrix Synapse по умолчанию используется реляционная база данных с открытым кодом PostgreSQL. Приложение регистрируется как Application Service (служба приложений) в Matrix Synapse и взаимодействует с ним через Application Service API.

Все homeserver-серверы взаимодействует между собой через Server-Server API. Любое событие в федеративной комнате (отправка сообщения, реакция на сообщение, изменения названия комнаты, добавление нового пользователя и т.д.), произошедшее на одном сервере приложения, проходит через эту цепочку и записывается в базу другого сервера приложения.

Каждый аккаунт однозначно определяется логином и сервером, например, @alice:example.com. Комната определяется ее именем и сервером, например,

#room:example.com. Matrix Synapse не хранит данные адреса почты, номера телефона, но хранит необратимые хеш-функции и осуществляет поиск именно по ним.

4.8.1 Настройка режима федерации

Для взаимодействия на сетевом уровне необходимо открыть доступ по протоколу TCP портам для входящего трафика, приведенным в таблице 24.

Таблица 24 — Порты для входящего трафика

Параметр	Описание	Тип	Значение по умолчанию
squadus_federation_bridge_int_port	Бридж порт для сервиса **squadus**	int	3300
synapse_client_int_port	Клиентский порт сервиса Synapse	int	8008
synapse_federation_int_port	Порт федерации сервиса Synapse	int	8448

Для установки приложения в режиме федерации необходимо присвоить значения переменным, приведенным в таблице 25.

Таблица 25 — Переменные для настройки федераций

Параметр	Описание	Тип	Значение по умолчанию
squadus_postgresql_enabled	Включить установку базы данных PostgreSQL	bool	true
squadus_federation_enabled	Включить режим федерации Squadus	bool	true
synapse_ip_range_blacklist	Список запрещенных IP-адресов для приема сообщений	list	-
synapse_ip_range_whitelist	Список разрешенных IP-адресов для приема сообщений	list	-
synapse_federation_domain_whitelist	Список разрешенных доменов для приема сообщений. Пример: im.example.com	list	-

Предварительно, необходимо установить библиотеку `psycopg2` (см. раздел «Программные требования»). Данная библиотека используется для управления конфигурацией сервиса базы данных PostgreSQL.

Для применения настроек на ранее подготовленном стенде следует запустить команду:

```
ansible-playbook playbooks/squadus.yml --tags federation --diff
```

4.9 Настройка отправки приложений к логам

В приложении реализована функция отправки копий логов на сторонний сервер в уже существующей системе заказчика. Настройка функции выполняется в следующем порядке:

1. Для настройки подключения необходимо в файле `~install_squadus/group_vars/squadus/main.yml` указать значения переменным, перечисленным в таблице 26.

Таблица 26 — Настройка отправки логов на сторонний сервер

Наименование переменной	Тип	Значение	Описание
syslog_ng_external_collector_ hostname	str	-	IP-адрес или домен стороннего сервера сбора логов
syslog_ng_external_port_ remote	int	-	Порт стороннего сервера сбора логов
syslog_ng_external_tier_send_ remote	bool	true / false (по умолчанию)	Включение/отключение функции отправки логов на сторонний сервер

Примечание — при отсутствии перечисленных в таблице 27 переменных следует добавить их вручную в файле `~install_squadus/group_vars/squadus/main.yml`.

2. Для выбора сервиса необходимо в файле `~install_squadus/group_vars/squadus/main.yml` найти словарь `syslog_ng_services` и добавить переменную `external_collector:` со значением `true`.

Пример:

```
      syslog_ng_services:
...
  squadus_apps:
    argon: {}
    bohrium: {}
    boron: {}
...
    praseodymium: {}
    sodium: {}
    squadus:
      programm_name: "squadus-.*"
      external_collector: true
    squadus_jiratrigger: {}
    tennessine: {}
```

3. Для внесения изменений и запуска сервиса в существующей системе следует выполнить команду:

```
ansible-playbook playbooks/squadus.yml -t syslog_ng  --diff -i hosts.yml
```

4. Для внесения изменений и запуска сервиса в новой системе после изменений файла `~install_squadus/group_vars/squadus/main.yml` необходимо выполнить установку системы в соответствии с разделом «Запуск установки».

4.10 Добавление стороннего корневого сертификата

В ПО реализована функция добавления стороннего корневого сертификата (далее — CA). Под сторонними подразумеваются самоподписанные CA или CA, полученные от третьих лиц.

Для работы с функцией необходимо открыть файл `~install_squadus/group_vars/squadus/main.yml` и в директории `install_squadus` указать значения переменным, представленным в таблице 27.

Таблица 27 — Добавление стороннего корневого сертификата

Наименование переменной	Тип переменной	Значение	Описание
prosody_extra_ca_enabled	boolean	true / false (по умолчанию)	Включение/ отключение функции поддержки сторонних CA
prosody_extra_tls_ca_directory	string	"/path/to/directory"	Директория, содержащая сторонние CA файлы в формате *.crt

Примечание — при отсутствии перечисленных в таблице 28 переменных следует добавить их вручную в файл `~install_squadus/group_vars/squadus/main.yml`.

Внесение изменений в существующей системе выполняется для подсистемы prosody с помощью команды:

```
ansible-playbook playbooks/squadus.yml -t prosody --diff -i hosts.yml
```

Для внесения изменений и запуска сервиса в новой системе после изменений файла `~install_squadus/group_vars/squadus/main.yml` необходимо выполнить установку системы в соответствии с разделом «Запуск установки».

4.11 Настройка системы для работы более 1000 пользователей

Для использования приложения 1000 и более пользователей перед установкой следует добавить переменную `tennessine_status_debounce_time` в соответствии с разделом «Установка системы для работы более 1000 пользователей».

После установки приложения без переменной `tennessine_status_debounce_time` возможно настроить систему для работы более 1000 пользователей с помощью команды:

```
ansible-playbook playbooks/squadus.yml --tags tennessin --diff
```

4.12 Централизованная установка настольных приложений

Централизованная установка настольного приложения для всех пользователей компьютера выполняется администратором. Такой тип установки использует дополнительное ПО SCCM (System Center Configuration Manager). Пользователь SCCM с ролью «Администратор» обладает правами на создание пакетов установок и распространение их содержимого на рабочие станции пользователей.

ПО разворачивается с помощью установщика msi "для всех пользователей" посредством обычной установки, с аргументом MSIINSTALLPERUSER="".

Пример команды:

```
msiexec /i "SquadusMsiPath" /qn MSIINSTALLPERUSER=""
```