УТВЕРЖДАЮ
Генеральный директор
ООО «КОВОРКИНГ ПЛАТФОРМ»

_____ Османов Ю.Ю.

«15» апреля 2025 г.

ИНСТРУКЦИЯ ПО УДАЛЕННОМУ ДОСТУПУ К ИНФРАСТРУКТУРЕ

с развёрнутым экземпляром ПО

SaaS-решения CoWork

Москва 2025

Оглавление

Аннотация

Документ предназначен для проверки автоматизированной системы CoWork, предназначенной для аудио–видеозвонков и онлайн конференций Экспертным советом при Минцифры России в рамках процедур по включению в Единый реестр ПО. Докупент составлен в полном соответствии с методическими рекомендациями АНО "ЦКИТ"

Документ содержит информацию об инфраструктуре с развёрнутым экземпляром ПО CoWork, являющимся SaaS-решением и публикуется на сайте Продукта https://co-work.ru Для части проверок необходимо иметь ключи доступа, которые можно получить у службы технической поддержки, контакты которой приведены в разделе 4 настоящего документа.

1. Общие сведения

Вся инфраструктура расположена на серверах, арендуемых правообладателем (ООО «КОВОРКИНГ ПЛАТФОРМ») у ООО «ВК» зарегистрированного по адресу РФ, 125167 г. Москва, Ленинградский проспект, д. 39, стр. 79, функционирующих в ЦОД по адресам:

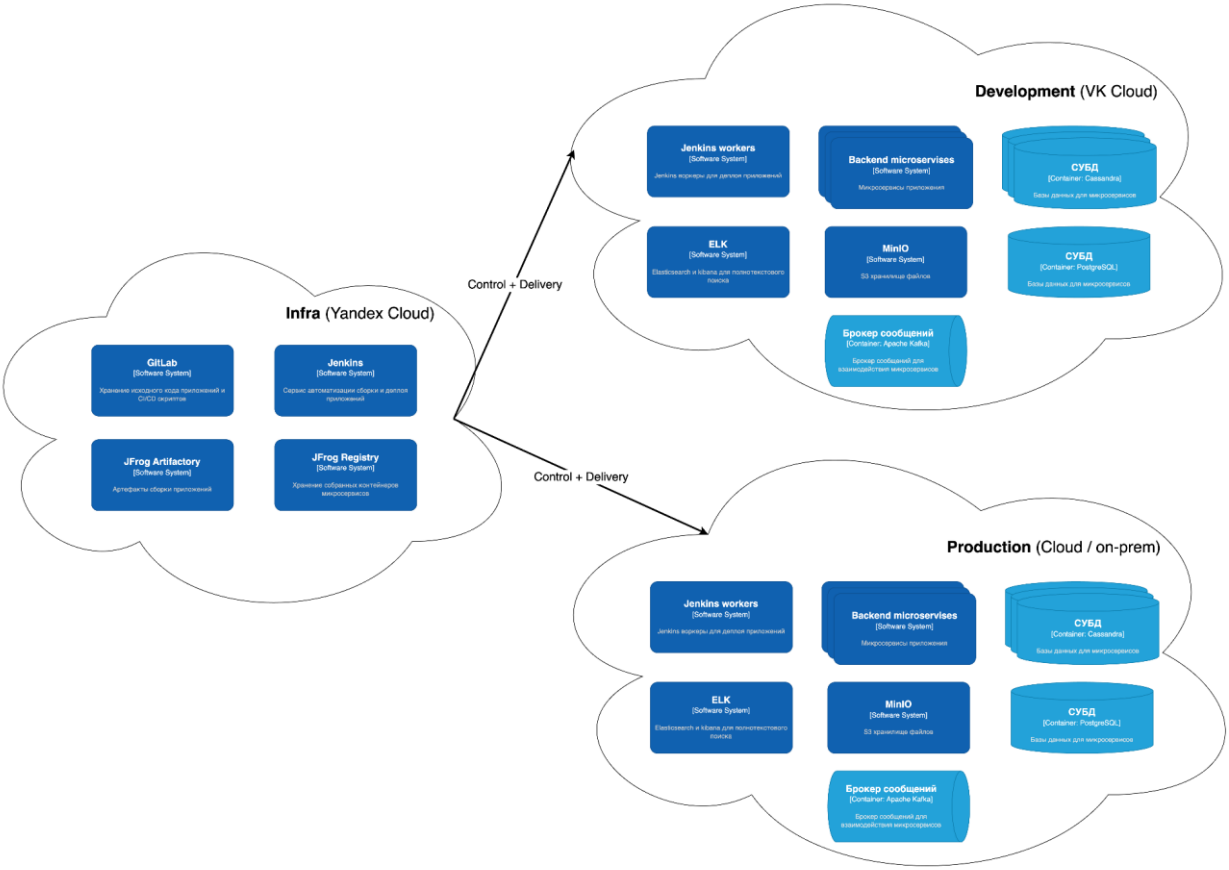
- РФ, г. Москва, проспект Мира, 105, стр. 6.
- РФ, г. Москва, Коровинское ш., 41.
- РФ, г. Москва, Чермянская ул., д. 4.

и у ООО «Яндекс.Облако», зарегистрированного по адресу РФ, 119021, г. Москва, ул. Льва Толстого, д. 16 пом. 528, функционирующих в ЦОД по адресам:

- РФ, г. Владимир, мкр. Энергетик, ул. Поисковая 1 к. 2;
- РФ, г. Сасово, ул. Пушкина 21;
- РФ, г. Калуга, 1-й Автомобильный пр-д 8.

2. Описание инфраструктуры

2.1.Общее описание



Данная схема демонстрирует принцип работы инфраструктуры продукта CoWork. В данный момент времени развернуто только один экземпляр ПО, расположенный в VK Cloud. Другие экземпляры ПО, отмеченные на схеме Production (Cloud/ on-prem) на данный момент времени недоступны и показаны для демонстрации принципа взаимодействия площадок Yandex Cloud и VK Cloud.

2.2.Процессы подготовки и выпуска ПО

2.2.1.Процессы CI/CD

Сборка всех платформ происходит с помощью Jenkins. Названия job:

- мобильного приложения: CW-IOS/b2b-ios
- Backend и frontend: набор job в папке CW-BACK

2.2.1.1.Сборка клиентов мобильного приложения

Аргументы для запуска сборки приложения:

- BRANCH - ветка в репозитории из которой будет выполнена сборка проекта;
- TAG - сборка коммита с выставленным тегом из выбранной выше ветки;
- ENVIRONMENT - выбор окружения (продакшн, пре-продакшн, стейдж);
- SERVER_HOST - возможность задать конкретный эндпоинт сервера для сборки приложения;
- TEST_FLIGHT - указывает необходимость отправки билда в AppStore.

При старте сборки, скрипт выполняет следующие шаги:

- Переключение на запрошенную ветку;
- Производит проверку формата кода по заданным параметрам;
- Инкрементирует текущий номер билда с коммитом;
- Производит сборку с указанными параметрами через скрипт Fastfile в корне проекта;
- После процесса сборки, готовый файл отправляется в Apple.

2.2.1.2.Сборка backend-микросервисов и frontend

Под каждый микросервис создана своя job сборки, которая подтягивает все зависимости из проекта.
Сборка параметризованная производится в Jenkins:

- открывается нужный пайплайн;
- выбирается окружение;
- выбирается проект;
- выбирается ветка;
- во вкладке Configure можно увидеть конечный скрипт сборки;
- собрать билд.

При сборке билда происходит подключение к репозиторию, скачивание ветки, далее через Gradle происходит сборка и автоматическая публикация собранного контейнера на сервере.
Аналогичная логика для всех имеющихся микросервисов проекта.

2.2.2.Подготовка к релизу

Определяется и согласовывается дата релиза.
Команды проводят ревью задач, уже слитых в основную ветку разработки (develop), проверяя результаты тестирования и готовность к релизу.
Анализируются задачи, находящиеся в работе:

- Если задача завершена и протестирована, включается в релиз.
- Если есть риски, обсуждается перенос на следующий релиз.
- Консультируются заинтересованные стороны (продукт, QA, маркетинг).

2.2.3.Выпуск iOS-приложения

2.2.3.1.Формирование релизной ветки

- От основной ветки разработки (develop) создаётся релизная ветка (release/x.y.z).
- В релизной ветке обновляется версия приложения.
- Все ожидаемые и подтвержденные к релизу задачи вливаются в релизную ветку.
- Собирается релизный билд и отправляется в TestFlight для регрессионного тестирования.

2.2.3.2.Подготовка к публикации

- Отдел маркетинга получает описание задач, ссылки на тикеты и составляет **release notes** для App Store.
- После завершения подготовки release notes создаётся новая версия приложения в App Store Connect.
- Если все задачи релиза проверены и подтверждены, билд отправляется на ревью в Apple.

- Окончание регресса не блокирует отправку билда, но финальный релиз зависит от его результата.

2.2.3.3.Публикация релиза

- После успешного завершения регрессионного тестирования и одобрения от Apple принимается финальное решение о релизе.
- Приложение выпускается **одновременно для всей аудитории**.
- Проводится смоук-тестирование на **production-сервере.**

2.2.3.4.Мониторинг и возможный хотфикс

- В течение нескольких дней после релиза проводится мониторинг:
  - Крашлитика.
  - Отзывы пользователей в App Store.
- Если выявлены критические баги:
  - В экстренных случаях готовится хотфикс, создается ветка (hotfix/x.y.z+1), исправления тестируются и оперативно публикуются.

2.2.3.5.Завершение релиза

- Если серьёзных проблем не выявлено, через **1-2 недели** релизная ветка (release/x.y.z) вливается в master и develop для синхронизации изменений.

2.2.4.Выпуск Android-приложения

2.2.4.1.Формирование релизной ветки

- От основной ветки разработки (develop) создаётся релизная ветка (release/x.y.z).
- В релизной ветке обновляется версия приложения (versionCode и versionName в build.gradle).
- Все ожидаемые и подтвержденные к релизу задачи вливаются в релизную ветку.
- Собирается **релизный APK** и передаётся команде тестирования для проведения регрессионного тестирования.

2.2.4.2.Подготовка к публикации

- Отдел маркетинга получает описание задач, ссылки на тикеты и составляет **release notes** для Google Play и RuStore.
- После завершения подготовки release notes создаётся новая версия приложения в **Google Play Console** и **RuStore Developer Console**.
- Если все задачи релиза проверены и подтверждены, финальный релизный билд подписывается и загружается в Google Play и RuStore.

2.2.4.3.Публикация релиза

- После успешного завершения регрессионного тестирования принимается финальное решение о релизе.

- Приложение публикуется **одновременно для всей аудитории**.
- Проводится смоук-тестирование на **production-сервере.**

2.2.4.4.Мониторинг и возможный хотфикс

- В течение нескольких дней после релиза проводится мониторинг:
    - Крашлитика.
- Если выявлены критические баги:
    - В экстренных случаях готовится хотфикс, создается ветка (hotfix/x.y.z+1), исправления тестируются и оперативно публикуются.

2.2.4.5.Завершение релиза

- Если серьёзных проблем не выявлено, через **1-2 недели** релизная ветка (release/x.y.z) вливается в master и develop для синхронизации изменений.

2.2.5.Выпуск Web-клиента

2.2.5.1.Формирование релизной ветки

- От основной ветки разработки (develop) создаётся релизная ветка (release/x.y.z).
- В релизной ветке обновляется версия приложения (в package.json).
- Все ожидаемые и подтвержденные к релизу задачи вливаются в релизную ветку.
- Собирается **релизный билд (статические файлы)** и разворачивается на **стейджинг-среде** для регрессионного тестирования.

2.2.5.2.Подготовка к публикации

- Отдел маркетинга получает описание задач, ссылки на тикеты и составляет **release notes**.
- Если все задачи релиза проверены и подтверждены, билд загружается в **Artifactory**.
- Готовится Docker-образ с web-приложением и отправляется в **Registry**.

2.2.5.3.Публикация релиза

- После успешного завершения регрессионного тестирования принимается финальное решение о релизе.
- Приложение деплоится на **production-серверы** через CI/CD.
- Проводится смоук-тестирование на **production-сервере.**

2.2.5.4.Мониторинг и возможный хотфикс

- В течение нескольких дней после релиза проводится мониторинг:
    - Логирование и метрики.
    - Пользовательские отзывы и аналитика.
- Если выявлены критические баги:
    - Готовится хотфикс, создается ветка (hotfix/x.y.z+1), исправления тестируются и оперативно деплоятся.

2.2.5.5.Завершение релиза

- Если серьёзных проблем не выявлено, через **1-2 недели** релизная ветка (release/x.y.z) вливается в master и develop для синхронизации изменений.

2.2.6.Выпуск backend-микросервисов

2.2.6.1.Формирование релизной ветки

- От основной ветки разработки (develop) создаётся релизная ветка (release/x.y.z).
- В релизной ветке обновляется версия микросервиса(ов).
- Все ожидаемые и подтвержденные к релизу задачи вливаются в релизную ветку.
- Собирается Docker-образ микросервиса(ов) и загружается в **jFrog Registry**.
- Готовятся скрипты для обновления сервиса (**helm или docker compose**).
- Разворачивается staging-версия сервиса(ов).

2.2.6.2.Подготовка к публикации

- Если все задачи релиза проверены и подтверждены, создаётся финальный релизный Docker-образ и загружается в **jFrog Registry**.

2.2.6.3.Публикация релиза

- После успешного завершения тестирования принимается финальное решение о релизе.
- Обновляется production-кластер (**helm или docker compose**).

2.2.6.4.Мониторинг и возможный хотфикс

- В течение нескольких дней после релиза проводится мониторинг:
  - Логирование и метрики.
- Если выявлены критические баги:
  - Готовится хотфикс, создается ветка (hotfix/x.y.z+1), исправления тестируются и оперативно деплоятся.

2.2.6.5.Завершение релиза

Если серьёзных проблем не выявлено, через **1-2 недели** релизная ветка (release/x.y.z) вливается в master и develop для синхронизации изменений.

2.3.Описание расположения файлов ПО

После подключения к инфраструктуре эксперт может ознакомиться со структурой и содержимым репозиториев. В проекте представлены следующие основные репозитарии

Репозиторий	Категория	Описание
bff	Backend	сервис “единая точка входа“ клиент-серверного взаимодействия
users	Backend	сервис отвечающий за работу с сущностью пользователя приложения
chats	Backend	сервис отвечающий за работу чатов

Репозиторий	Категория	Описание
messages	Backend	сервис отвечающий за работу с сообщениями
calls	Backend	сервис звонков
gem-call	Backend	сервис звонков
upload	Backend	сервис отвечающий за работу с файлами в чатах
stickers	Backend	сервис стикеров
realtime	Backend	сервис отвечающий за транспорт между сервисами
notifications	Backend	сервис отправки уведомлений
deeplinks	Backend	сервис для работы с диплинками
search	Backend	содержит файлы сервиса для функционирования поиска в системе
auth	Backend	сервис аутентификации пользователей
workspaces	Backend	содержит файлы сервиса организаций
infra	Backend	сервис содержащий файлы для конфигурации инфраструктуры
gem-ios	iOS	Основной репозиторий приложения
gem-ios-haring	iOS	Fork внешней библиотеки для Markdown рендеринга на iOS
gem-ios-jungo	iOS	Fork внешней библиотеки системных функций
gem-ios-layout-kit	iOS	Fork внешней библиотеки быстрого просмотра макетов для iOS
gem-ios-lottie	iOS	Fork внешней библиотеки для воспроизведения анимаций
gem-ios-network-contract	iOS	Репозиторий контрактов с backend
cw-infra	Инфраструктура	Скрипты развертывания инфраструктурных сервисов
cw-infra-apps	Инфраструктура	Скрипты сборки и деплоя сервисов
cw-jenkins	Инфраструктура	Скрипты Jenkins jobs

2.4.Исходные коды мобильных приложений

Исходные коды мобильных приложений доступны в соответствующих репозитариях, описанных в разделе 2.3 «Описание расположения файлов ПО»

2.5.Описание микроконтейнеров

Имя микроконтейнера	Описание
message-app	контейнер сервиса сообщений отвечающий за работу с сообщениями в приложениями
gem-call-events-handler-app	интеграция эвентов от call-event-handler-app в мессенджер, например отправка сообщений о записи звонка
call-event-handler-app	контейнер сервиса для обработки асинхронных действий пользователей на звонке, например подключение/отключение со звонка
gem-call-app	контейнер сервиса интеграции call-app и call-realtime-app в мессенджер
call-app	контейнер для работы с апи звонков, отвечает за запуск/окончание/получение звонков
call-realtime-app	контейнер для прослушивания реалтайм обновлений на звонке

Имя микроконтейнера	Описание
bff-app	контейнер сервиса bff который является точкой входа для клиент-серверного взаимодействия
realtime-app	контейнер сервиса отвечающего за рассылку событий происходящих в реальном времени в приложении
message-event-handler-app	контейнер сервиса для обработки событий связанных с сообщениями
redis	контейнер кэш сервиса Redis
chat-app	контейнер сервиса чатов который отвечает за работу чатов в приложении
cassandra-sticker-service	контейнер базы данных сервиса по работе со стикерами
redis-sticker-service	контейнер кэш сервиса по работе со стикерами
notification-app	контейнер сервиса уведомлений. Отвечает за рассылку уведомлений на разные платформы
message-link-preview-handler-app	контейнер сервиса для обработки событий связанных с link-preview
deeplink-app	контейнера сервиса по работе с диплинк
regular-chat-splitter-app	контейнер сервиса подготавливающего сообщения для сервиса уведомлений
voip-splitter-app	контейнер сервиса подготавливающего voip сообщения для сервиса уведомлений
desktop-app	контейнер сервиса использующегося сервисом уведомлений для отправки уведомлений на десктопное приложение
web-app	контейнер сервиса используещегося сервисом уведомлений для отправки уведомлений на браузер Chrome
android-app	контейнер сервиса использующегося сервисом уведомлений для отправки уведомлений на устройства android
chat-event-handler-app	контейнер сервиса для обработки событий связанных с чатами
ios-app	контейнер сервиса использующегося сервисом уведомлений для отправки уведомлений на устройства ios
sticker-app	контейнер сервиса отвечающего за работу со стикерами в приложении
search-app	контейнер сервиса обеспечивающего поиск в приложении по разным критериям пользователи/сообщения и тд
repartitioner-app	контейнер сервиса разделяющего очередь сообщений по приоритетности обработки
postgres-notifications	контейнер базы данных использующийся для сервиса уведомлений
postgres-search	контейнер базы данных использующийся для сервиса поиска
big-chat-splitter-app	контейнер сервиса разделяющего очередь сообщений по приоритетности обработки
upload-app	контейнер сервиса работы с файлами
notification-messages-	контенер преобразования

Имя микроконтейнера	Описание
transformer-app	
user-app	контейнер сервиса пользователей который аггрегирует работу приложения с сущностью пользователя
kafka-init	контейнер сервиса брокера сообщений Apache Kafka
redis-users	контейнер кэш сервиса пользователей
livekit-webhook-handler-app	контейнер сервиса обработчика вебхуков для сервиса звонков
notification-unregister-tokens-app	контейнер сервиса управления токенами для рассылки уведомлений
auth-app	контейнер сервиса аутентификации отвечающий за проверку прав доступа пользователей
huawei-app	контейнер сервиса использующегося сервисом уведомлений для отправки уведомлений на устройства huawei
safari-app	контейнер сервиса использующегося сервисом уведомлений для отправки уведомлений на браузеры safari
kafka-0	контейнер сервиса брокера сообщений Apache Kafka
kafka-1	контейнер сервиса брокера сообщений Apache Kafka
kafka-2	контейнер сервиса брокера сообщений Apache Kafka
kafka-ui	контейнер UI сервиса брокера сообщений Apache Kafka
workspace-app	контейнер сервиса workspace. Отвечает за информацию и данных об текущей организации пользователе
kibana_search	контейнер сервиса Kibana используется для полнотекстового поиска в приложении
elastic_search	контейнер сериса Elasticsearch используется для полнотекстового поиска в приложении
auth-cassandra	контейнер базы данных сервиса auth
users-cassandra	контейнер базы данных сервиса users
messages-cassandra	контейнер базы данных сервиса messages
chats-cassandra	контейнер базы данных сервиса chats
calls-cassandra	контейнер базы данных сервиса calls
uploads-cassandra	контейнер базы данных сервиса uploads
notifications-cassandra	контейнер базы данных сервиса notifications
workspaces-cassandra	контейнер базы данных сервиса workspaces
minio01	контейнер хранилища файлов s3
livekit	контейнер сервиса звонков (ядро)
mon01	контейнер сервиса мониторинга приложения

3. Подключение к инфраструктуре

3.1.Требования к окружению

- Автоматизированное рабочее место под управлением операционной системы Linux/Windows 10 и выше/MacOS 13 и выше.
- Отсутствие блокировок к облачным ресурсам Yandex Cloud московского региона автономная сеть интернет AS200350 (158.160.0.0/16).
- Разрешение имени wireguard-03.co-work.ru. Это можно сделать например, путем путем подключения к данному адресу используя браузер; факт открытия страницы в браузере будет являться подтверждением успешного разрешения имени, выполнять инструкции на данной странице, содержащий стандартный текст для продукта Wireguard, не требуется.
- Установленный клиент wireguard с сайта https://www.wireguard.com/install/ для используемой операционной системы.
- Браузер Google chrome актуальной версии.

3.2.Подключение к зоне разработки

- Связаться со специалистом технической поддержки и получить
  - wireguard конфигурацию
  - Учетная запись для авторизации
- В настройках клиента wireguard требуется добавить конфигурацию подключения путем импортирования предоставленного файла конфигурации.
- После добавления конфигурации произвести подключение к удаленной сети CoWork используя клиент wireguard.
- Проверить доступность ресурсов удаленной сети путем подключения к https://jenkins.ii-p001.local используя браузер. В процессе подключения будет выдана ошибка проверки сертификата, требуется выбрать “продолжить подключение”.

3.3.Ресурсы в зоне разработки

- https://gitlab.ii-p001.local/ - основное хранилище кода.
- https://jenkins.ii-p001.local - инструмент автоматизации разработки и сборки приложения.
- https://registry.ii-p001.local - хранилище собранных образов.

4. Контакты технических специалистов

Для авторизации и аутентификации на указанных ресурсах требуется использовать предоставленные учетные записи.

В связи с тем, что программный продукт представляет собой SaaS решение и развернут в единственном экземпляре на серверах ООО «КОВОРКИНГ ПЛАТФОРМ», подключение к инфраструктуре возможно только с помочью технического специалиста компании ООО «КОВОРКИНГ ПЛАТФОРМ»

Для организации доступа и проведения демонстрации необходимо связаться с одним из технических специалистов в рабочие дни с 11 до 17:00 по московскому времени

ФИО	**email**	**Telegram**
Леонид Сокуров*	leonid.sokurov@co-work.ru	https://t.me/LeonidST
Иван Шелапутов	ivan.shelaputov@co-work.ru	https://t.me/shelaputov

*) Основной контакт, предпочтительно связываться с ним