

# Cinimex Integration Framework

## Руководство администратора

Версия 1.0.0

АО «СИНИМЕКС-ИНФОРМАТИКА»

## 2 ОГЛАВЛЕНИЕ

---

1	Глоссарий.....	3
2	Введение .....	3
2.1	Цели документа .....	3
2.2	Описание ПО .....	3
2.3	Целевая аудитория и уровень подготовки .....	4
2.4	Связанные документы .....	4
3	Подготовка к работе .....	5
3.1	Требования к рабочей станции.....	5
3.1.1	Требования к техническому обеспечению .....	5
3.1.2	Требования к программному обеспечению .....	5
3.1.3	Требования к доступам .....	5
3.2	Порядок загрузки дистрибутива .....	6
3.3	Комплектность поставки .....	7
3.4	Настройка рабочего места .....	7
3.5	Установка дистрибутива на рабочую станцию .....	8
4	Установка фреймворка на платформе Cinimex Microservice Platform .....	8
4.1	Создание технологической учетной записи .....	8
4.2	Создание группы проектов в GITLAB .....	12
4.2.1	Создание группы для компонентов фреймворка.....	12
4.2.2	Создание группы интеграционных приложений.....	13
4.3	Добавление технологической учетной записи в группы .....	14
4.4	Настройка переменных CI&CD.....	14
4.4.1	Настройка переменных CI&CD для группы CMXAIPFWK.....	14
4.4.2	Настройка переменных CI&CD для группы интеграционных приложений.....	16
4.5	Сборка и публикация проектов .....	16
4.5.1	Сборка и публикация проекта msfwk-image-ci .....	16
4.5.2	Сборка и публикация проекта msfwk .....	17
4.5.3	Сборка проекта msfwk-server-config.....	17
4.5.4	Сборка и публикация проекта msfwk-server-hawtio .....	17
5	Создание и публикация проекта рабочей области (namespace).....	18
5.1	Создание токена для доступа к реестру образов.....	18
5.2	Создание проекта рабочей области .....	19
5.2.1	Сборка maven-archetype проекта .....	19
5.2.2	Генерация проекта рабочей области .....	19
5.2.3	Публикация проекта рабочей области в GitLab.....	20

# 1 ГЛОССАРИЙ

---

ПО	Программное обеспечение
ОС	Операционная система
CI&CD	Continues Integration, Continues Delivery and Continues Deployment - процессы непрерывной интеграции, доставки и развертывания ПО
Java	Популярный типизированный объектно-ориентированный язык программирования
JVM	от англ. Java Virtual Machine, виртуальная машина Java — основная часть среды исполнения Java программ
JRE	от англ. Java Runtime Environment - среда исполнения для Java приложений, включающая в себя виртуальную машину Java (JVM) и набор базовых библиотек
JDK	от англ. Java Development Kit - расширенный комплект разработчика приложений на языке Java, включающий в себя компилятор, среду исполнения java программ (JRE), набор дополнительных утилит и инструментов отладки, примеры и документацию
ZIP	Популярный формат архивации файлов и сжатия данных без потерь
Maven archetype	Специальный тип проекта maven, описывающий шаблон типового проекта и позволяющий генерировать проекты из шаблона путем выполнения одной параметризованной команды
HDD	Накопитель на жёстких магнитных дисках (от англ. hard (magnetic) disk drive)
SSD	Твердотельный накопитель (от англ. solid-state drive)

# 2 ВВЕДЕНИЕ

---

## 2.1 ЦЕЛИ ДОКУМЕНТА

Настоящий документ представляет из себя руководство администратора по установке и настройке компонентов фреймворка Cinimex Integration Framework версии 1.0.0 на платформе Cinimex Microservice Platform.

## 2.2 ОПИСАНИЕ ПО

**Полное наименование программы для ЭВМ:** Cinimex Integration Framework.

**Разработчик:** Акционерное общество «СИНИМЕКС-ИНФОРМАТИКА»

ОГРН 1027700553781

ИНН / КПП 773 414 8839 / 770 501 001

115184, г. Москва, улица Большая Татарская, д. 35, стр. 3

+7 495 955-7-955

tender@cinimex.ru

**Область назначения:** Программа предназначена для разработки интеграционных приложений, автоматизирующих процессы обмена электронными документами (далее - ЭД) между автоматизированными системами (далее - АС).

**Область применения:** Программа применяется при автоматизации процессов электронного документооборота между АС.

### **Функциональные возможности программы:**

- 1) Реализация транспортных протоколов обмена данными;
- 2) Преобразование протоколов обмена данными;
- 3) Преобразование форматов ЭД: XML, JSON, CSV и другие текстовые, двоичные и отраслевые форматы;
- 4) Управление процессом обработки ЭД (маршрутизация, рассылка, фильтрация, агрегация, обогащение и произвольная алгоритмическая обработка);
- 5) Мониторинг здоровья и производительности процессов обмена ЭД;
- 6) Журналирование событий аудита и обработки ЭД;
- 7) Управление процессами сохранения ЭД в системах управления базами данных.

**Язык программирования:** Java.

### 2.3 ЦЕЛЕВАЯ АУДИТОРИЯ И УРОВЕНЬ ПОДГОТОВКИ

Документ предназначен в первую очередь для администраторов.

Предполагается, что администратор платформы владеет как минимум следующим набором базовых знаний и навыков:

- Уверенное использование ОС Linux;
- Умение работать с системой контроля версий git;
- Знакомство с веб-интерфейсом системы GitLab;
- Понимание основ работы платформы Kubernetes.

### 2.4 СВЯЗАННЫЕ ДОКУМЕНТЫ

Помимо настоящего руководства, администратору рекомендуется ознакомиться со следующим списком документов из архива эксплуатационной документации:

- Cinimex Integration Framework. Реферат.pdf, версия 1.0.0  
Документ содержит краткую информацию о функциональных возможностях фреймворка и области его применения.
- Cinimex Integration Framework. Руководство разработчика.pdf, версия 1.0.0  
В документе отражена инструкция по разработке и сборке приложений с использованием фреймворка.

## 3 ПОДГОТОВКА К РАБОТЕ

### 3.1 ТРЕБОВАНИЯ К РАБОЧЕЙ СТАНЦИИ

#### 3.1.1 Требования к техническому обеспечению

В таблице ниже приведена минимальная и рекомендуемая конфигурация рабочей станции разработчика.

Компонент	Минимальная конфигурация	Рекомендуемая конфигурация
Экран	Разрешение 1024x768	Разрешение 1920x1080
Процессор	Современный, минимум 1 ядро	Современный, минимум 4 ядра
Оперативная память	Минимум 2 ГБ	Минимум 8 ГБ
Жесткий диск	Минимум 1 ГБ свободного места, HDD	Минимум 2 ГБ свободного места, SSD
Видеоадаптер	Интегрированный или дискретный	Интегрированный или дискретный
Сетевая плата	10 Мбит/с	100 Мбит/с
Доступ в сеть Интернет (для варианта установки онлайн)	Скорость входящего соединения минимум 1 Мбит/с	Скорость входящего соединения минимум 10 Мбит/с
Дополнительное оборудование	Клавиатура, мышь или аналоги	Клавиатура, мышь или аналоги

#### 3.1.2 Требования к программному обеспечению

На рабочей станции администратора должно быть установлено следующее ПО.

ПО	Описание	Версия	Команда для проверки установленной версии
ОС	Операционная система: <ul style="list-style-type: none"><li>64 битная версия Microsoft Windows 10, 8, 7 (SP1)</li><li>macOs &gt;= 10.11</li><li>Linux десктоп версии (Ubuntu, Debian, CentOS, openSUSE, RHEL, Oracle Linux)</li></ul>		-
JDK	Расширенный комплект разработчика приложений на языке Java, включающий в себя компилятор, среду исполнения java программ (JRE), набор дополнительных утилит и инструментов отладки, примеры и документацию	8.x (1.8.x)	java -version
git	Распределённая система версионирования и управления исходными кодами. <i>При установке на ОС Windows:</i> Git for Windows с эмулятором bash	2.x	git version
Maven	Система сборки Java приложений.	3.x	mvn -version

#### 3.1.3 Требования к доступам

Установка компонентов фреймворка на платформе Cinimex Microservice Platform осуществляется с рабочей станции администратора, удовлетворяющей пунктам 3.1.2 и 3.1.3 данного руководства.

Помимо соответствия требованиям к техническим средствам и ПО, администратору фреймворка должны быть предоставлены следующие доступы к серверам платформы Cinimex Microservice Platform:

- сетевой доступ к серверу GitLab;
- сетевой доступ к серверу GitLab Container Registry;
- учётная запись для работы в веб-интерфейсе сервера GitLab с правами на создание группы проектов (у учетной записи должен быть установлен признак «Can create group»).

Для получения перечисленных доступов требуется обратиться с соответствующим запросом к администратору платформы Cinimex Microservice Platform.

Далее по документу вместо реальных значений параметров подключения к платформе будут использоваться значения для подстановки:

Значение для подстановки	Описание	Пример
<GitLab DOMAIN>	Доменное имя сервера GitLab	gitlab.infra.k8sdemo.vm.cmx.ru
<GitLab URL>	URL сервера GitLab	https://gitlab.infra.k8sdemo.vm.cmx.ru
<Registry DOMAIN>	Доменное имя сервера GitLab Container Registry	registry.infra.k8sdemo.vm.cmx.ru
<Registry URL>	URL сервера GitLab Container Registry	https://registry.infra.k8sdemo.vm.cmx.ru
<Platform DOMAIN>	Базовый домен кластера платформы	k8sdemo.vm.cmx.ru
<GitLab USER>	Имя пользователя для доступа в GitLab	msfwk-admin
<GitLab EMAIL>	Email адрес, используемый при создании учетной записи администратора в GitLab	msfwk-admin@domain
<Platform CA>	Корневой сертификат (CA) сертификационного сервера, которым подписаны все сертификаты на платформе Cinimex Microservice Platform	-----BEGIN CERTIFICATE----- MIIDEzCCAfugAwIBAgIJAIiz6ML/HZo3MA 0GCSqGSIb3DQEBCwUAMCAxHjAcBgNVBAMM FWluZnJhLmNteGFzLnZtLmNteC5ydTAeFw 0xOTA0MTUxMDA5NTFaFw0yOTA0 ... -----END CERTIFICATE-----

Перед выполнением команд инструкции требуется заменить все значения для подстановки реальными данными.

### 3.2 ПОРЯДОК ЗАГРУЗКИ ДИСТРИБУТИВА

Дистрибутив с ПО Cinimex Integration Framework поставляется в электронном виде в формате ZIP-архива cinimex-integration-framework.v1.0.0.zip, содержащего в себе исходные коды и документацию.

Загрузка дистрибутива осуществляется с официального сайта компании изготовителя:  
<https://www.cinimex.ru/solutions/integration-framework/>

Так же возможен вариант передачи дистрибутива заказчику напрямую. Ключ лицензии и пароль для доступа к странице загрузки архива с дистрибутивом передаются отдельно.

В случае отсутствия или ограниченного доступа к сети Интернет с рабочего места администратора, возможен вариант офлайн установки. Для этого со страницы фреймворка потребуется дополнительно скачать и переместить на целевую рабочую станцию администратора

архив offline.v1.0.0.zip, содержащий в себе скомпилированные модули фреймворка и все зависимые библиотеки, необходимые для сборки приложений.

Офлайн вариант поставки может также использоваться в целях ускорения процесса подготовки рабочей станции к работе.

### 3.3 КОМПЛЕКТНОСТЬ ПОСТАВКИ

Архив cinimex-integration-framework.v1.0.0.zip, состоит из следующих каталогов:

Папка	Описание																				
src/	Папка с репозиториями исходных кодов модулей фреймворка: <table border="1"><thead><tr><th></th><th></th></tr></thead><tbody><tr><td>/msfwk</td><td>исходные коды программных модулей фреймворка</td></tr><tr><td>/msfwk-template-svc</td><td>исходный код maven archetype проекта с шаблоном интеграционного сервиса</td></tr><tr><td>/msfwk-template-namespace</td><td>исходный код maven archetype проекта с шаблоном общей рабочей области (namespace) в среде развертывания Cinimex Microservice Platform</td></tr><tr><td>/msfwk-examples</td><td>исходные коды примеров интеграционных приложений с использованием фреймворка</td></tr><tr><td>/msfwk-image-ci</td><td>исходный код образа ci&amp;cd контейнера</td></tr><tr><td>/msfwk-image-amq</td><td>исходный код образа контейнера для локального запуска ActiveMQ Artemis</td></tr><tr><td>/msfwk-server-config</td><td>исходный код сервера для централизованного управления конфигурацией</td></tr><tr><td>/msfwk-server-hawtio</td><td>исходный код веб-консоли для управления интеграционными приложениями</td></tr><tr><td>/msfwk-env-docker</td><td>исходный код для запуска среды исполнения docker</td></tr></tbody></table>			/msfwk	исходные коды программных модулей фреймворка	/msfwk-template-svc	исходный код maven archetype проекта с шаблоном интеграционного сервиса	/msfwk-template-namespace	исходный код maven archetype проекта с шаблоном общей рабочей области (namespace) в среде развертывания Cinimex Microservice Platform	/msfwk-examples	исходные коды примеров интеграционных приложений с использованием фреймворка	/msfwk-image-ci	исходный код образа ci&cd контейнера	/msfwk-image-amq	исходный код образа контейнера для локального запуска ActiveMQ Artemis	/msfwk-server-config	исходный код сервера для централизованного управления конфигурацией	/msfwk-server-hawtio	исходный код веб-консоли для управления интеграционными приложениями	/msfwk-env-docker	исходный код для запуска среды исполнения docker
/msfwk	исходные коды программных модулей фреймворка																				
/msfwk-template-svc	исходный код maven archetype проекта с шаблоном интеграционного сервиса																				
/msfwk-template-namespace	исходный код maven archetype проекта с шаблоном общей рабочей области (namespace) в среде развертывания Cinimex Microservice Platform																				
/msfwk-examples	исходные коды примеров интеграционных приложений с использованием фреймворка																				
/msfwk-image-ci	исходный код образа ci&cd контейнера																				
/msfwk-image-amq	исходный код образа контейнера для локального запуска ActiveMQ Artemis																				
/msfwk-server-config	исходный код сервера для централизованного управления конфигурацией																				
/msfwk-server-hawtio	исходный код веб-консоли для управления интеграционными приложениями																				
/msfwk-env-docker	исходный код для запуска среды исполнения docker																				
docs/	Эксплуатационная документация на фреймворк, включая настоящее руководство																				

### 3.4 НАСТРОЙКА РАБОЧЕГО МЕСТА

Первым делом требуется задать параметры учетной записи администратора фреймворка, для публикации проектов в системе GitLab. Для этого в командной строке требуется последовательно выполнить следующие команды:

```
git config --global user.email <GitLab EMAIL>
git config --global user.name <GitLab USER>
```

При использовании ОС Windows на рабочей станции администратора, требуется включить режим автоматической конвертации символов перевода конца строки при загрузке и публикации репозитория исходных кодов:

```
git config --global core.autocrlf true
```

Также требуется уточнить у администратора платформы Cinimex Microservice Platform тип сертификата, используемого для GitLab. Если GitLab настроен на использование самоподписанного сертификата, то требуется отключить проверку сертификатов для системы контроля версий git.

Для этого требуется выполнить следующую команду:

```
git config --global http.sslVerify false
```

### 3.5 УСТАНОВКА ДИСТРИБУТИВА НА РАБОЧУЮ СТАНЦИЮ

Установка фреймворка на рабочую станцию администратора выполняется в соответствии с пунктом 5.4 документа «Cinimex Integration Framework. Руководство разработчика.pdf» версии 1.0.0.

## 4 УСТАНОВКА ФРЕЙМВОРКА НА ПЛАТФОРМЕ CINIMEX MICROSERVICE PLATFORM

---

### 4.1 СОЗДАНИЕ ТЕХНОЛОГИЧЕСКОЙ УЧЕТНОЙ ЗАПИСИ

Перед загрузкой компонентов фреймворка в систему контроля версий GitLab требуется создать специальную технологическую учетную запись. Технологическая учетная запись - разновидность учетной записи, применяемая в случаях, когда пользователем выступает автоматизированная система.

Технологическая учетная запись используется для доступа к GitLab следующими компонентами фреймворка:

- Сервером конфигурации msfwk-server-config  
Настройки интеграционных приложений и их среды окружения хранятся в отдельном репозитории GitLab. Технологическая учетная запись используется для доступа к этому репозиторию.
- Скриптами CI&CD pipeline  
Технологическая учетная запись используется для создания специального secret-объекта, применяемого для загрузки образов контейнеров приложений в среду Kubernetes.

Для создания технологической учетной записи требуется выполнить следующие действия:

**Шаг 1.** Авторизоваться в веб-интерфейсе системы GitLab под своей учетной записью [<GitLab USER>](#)

Ссылка для доступа [<GitLab URL>](#) (пример: <https://gitlab.infra.k8sdemo.vm.cmx.ru>)

Sign in Register

Username or email  
msfwk-admin

Password  
.....

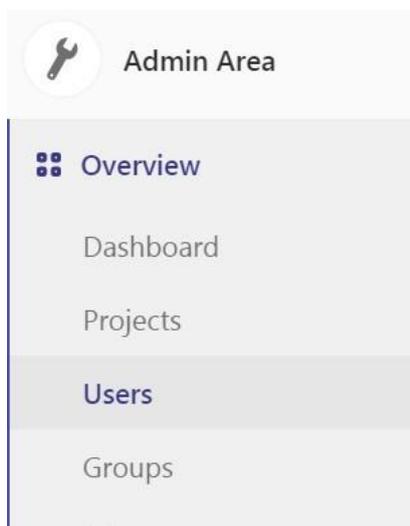
Remember me [Forgot your password?](#)

Sign in

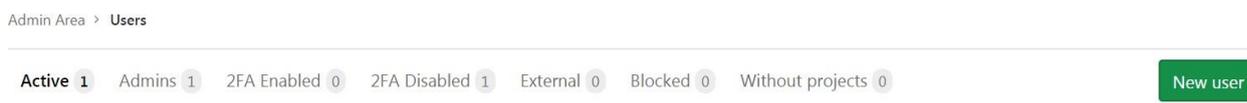
**Шаг 2.** Перейти в пункт меню «Admin area»



**Шаг 2.** Выбрать пункт «Users» в боковом меню



**Шаг 3.** Нажать на кнопке «New User»



**Шаг 4.** Заполнить обязательные поля формы.

В качестве имени учетной записи указать значение `msfwk-service-account` .

В качестве email указать специальный адрес электронной почты, используемый в компании для технологических учетных записей.

Если специального email-адреса в компании нет, а на сервере GitLab выключен режим обязательного подтверждения email (настройка по умолчанию), то можно указать любой несуществующий email в зарезервированной доменной зоне local. Например, подойдет значение msfwk-service-account@local .

**Шаг 5.** Выбрать тип учетной записи «Regular» и убедиться, что флажок «Can create group» установлен.

Admin Area > New User

## New user

### Account

Name	<input type="text" value="msfwk-service-account"/>
	* required
Username	<input type="text" value="msfwk-service-account"/>
	* required
Email	<input type="text" value="msfwk-service-account@local"/>
	* required

### Access

Projects limit	<input type="text" value="100000"/>
Can create group	<input checked="" type="checkbox"/>
Access level	<input checked="" type="radio"/> <b>Regular</b> Regular users have access to their groups and projects
	<input type="radio"/> <b>Admin</b> Administrators have access to all groups, projects and users and can
External	<input type="checkbox"/> External users cannot see internal or private projects unless access i

**Шаг 6.** Нажать кнопку «Create» в нижней части экрана.

**Шаг 7.** Убедиться, что учетная запись была успешно создана.

User was successfully created.

**Шаг 8.** Далее требуется выпустить обезличенный токен доступа.

Для этого необходимо:

- 1) перейти на вкладку «Impersonation Tokens»
- 2) заполнить поле «Name» значением msfwk-service-token
- 3) выставить флаги read\_repository и read\_registry
- 4) нажать кнопку «Create impersonation token».

## msfwk-service-account

Account Groups and projects SSH keys Identities **Impersonation Tokens**

### Add a impersonation token

Pick a name for the application, and we'll give you a unique impersonation token.

#### Name

#### Expires at

#### Scopes

- api**  
Grants complete read/write access to the API, including all groups and projects.
- read\_user**  
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, /users.
- sudo**  
Grants permission to perform API actions as any user in the system, when authenticated as an admin user.
- read\_repository**  
Grants read-only access to repositories on private projects using Git-over-HTTP (not using the API).
- read\_registry**  
Grants read-only access to container registry images on private projects.

Create impersonation token

**Шаг 9.** При успешном создании токена будет осуществлен автоматический переход на страницу с отображением его значения.

Требуется сохранить значение токена (подсвечено синим в примере вывода ниже) во временном файле для использования в последующих пунктах настоящей инструкции.

**Примечание:** При закрытии страницы у администратора не будет другой возможности получить значение токена. Если токен будет утерян до завершения установки, то потребуется выпустить новый токен и пройти незавершенные пункты инструкции заново.

Admin Area > Users > msfwk-service-account

A new impersonation token has been created.

msfwk-service-account Impersonate Edit

Account Groups and projects SSH keys Identities **Impersonation Tokens**

Your New Impersonation Token

CP5zZx4zzjB1c2XTxk3

Make sure you save it - you won't be able to access it again.

## 4.2 СОЗДАНИЕ ГРУППЫ ПРОЕКТОВ В GITLAB

### 4.2.1 Создание группы для компонентов фреймворка

Для установки компонентов фреймворка на платформу требуется создать отдельную группу проектов в GitLab с именем CMXAIPFWK. Для этого требуется выполнить следующие действия:

**Шаг 1.** Авторизоваться в веб-интерфейсе системы GitLab под своей учетной записью [<GitLab USER>](#)

Ссылка для доступа [<GitLab URL>](#) (пример: <https://gitlab.infra.k8sdemo.vm.cmx.ru>)

**Шаг 2.** Перейти на страницу создания новой группы через пункт «New group» выпадающего меню с пиктограммой + в главной строке меню:

GitLab Projects Groups Activity Milestones Snippets

Projects

Your projects 6 Starred projects 0 Explore projects

- New project
- New group
- New snippet

**Шаг 3.** Заполнить поле «Group name» значением CMXAIPFWK и нажать кнопку «Create group».

## New group

Groups allow you to manage and collaborate across multiple projects. Members of a group have access to all of its projects.

Groups can also be nested by creating subgroups.

Projects that belong to a group are prefixed with the group namespace. Existing projects may be moved into a group.

### Group name

### Group URL

### Group description (optional)

### Group avatar

 No file chosen

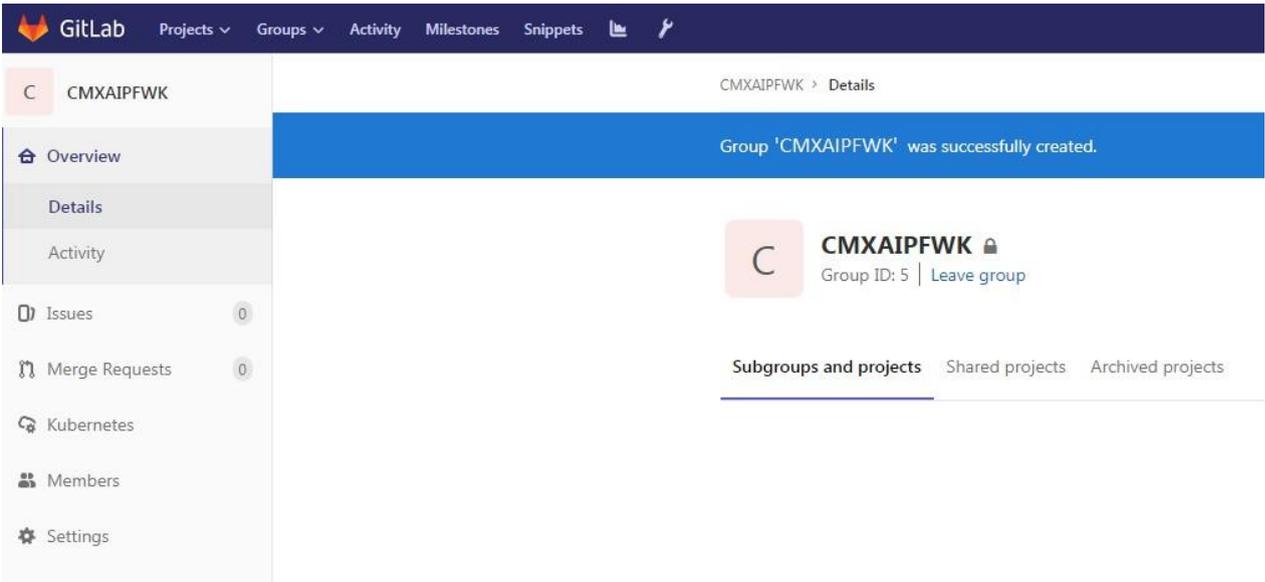
The maximum file size allowed is 200KB.

### Visibility level

Who will be able to see this group? [View the documentation](#)

-  Private  
The group and its projects can only be viewed by members.
-  Internal  
The group and any internal projects can be viewed by any logged in user.
-  Public  
The group and any public projects can be viewed without any authentication.

**Шаг 4.** В случае успеха на экран будет выведено соответствующее сообщение:



The screenshot shows the GitLab web interface. At the top, there is a navigation bar with the GitLab logo and menu items: Projects, Groups, Activity, Milestones, Snippets, and a search icon. Below the navigation bar, the breadcrumb path is 'CMXAIPFWK > Details'. A prominent blue banner displays the message: 'Group 'CMXAIPFWK' was successfully created.' Below this banner, the group details are shown: a group icon 'C', the name 'CMXAIPFWK' with a lock icon, and 'Group ID: 5 | Leave group'. At the bottom, there are tabs for 'Subgroups and projects', 'Shared projects', and 'Archived projects'. On the left side, a sidebar menu is visible with options: Overview, Details (selected), Activity, Issues (0), Merge Requests (0), Kubernetes, Members, and Settings.

### 4.2.2 Создание группы интеграционных приложений

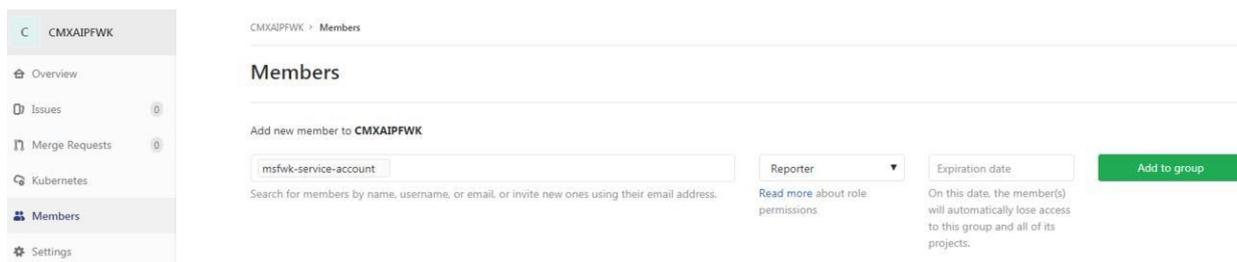
Требуется создать группу проектов GitLab для размещения репозиториев исходных кодов интеграционных приложений и их среды окружения.

Для этого требуется выполнить шаги пункта 4.2.1, но вместо имени CMXAIPFWK использовать другое название группы. Например, название MSFWK-EXAMPLES.

### 4.3 ДОБАВЛЕНИЕ ТЕХНОЛОГИЧЕСКОЙ УЧЕТНОЙ ЗАПИСИ В ГРУППЫ

Требуется добавить технологическую учетную запись msfwk-service-account в ранее созданные группы с ролью «Reporter».

Добавление учетных записей в группы осуществляется через раздел меню «Members» страницы группы. Для добавления записи необходимо заполнить поле «Search for a user» именем пользователя msfwk-service-account, поменять значение роли с Guest на Reporter в выпадающем меню и нажать кнопку «Add to group».



### 4.4 НАСТРОЙКА ПЕРЕМЕННЫХ CI&CD

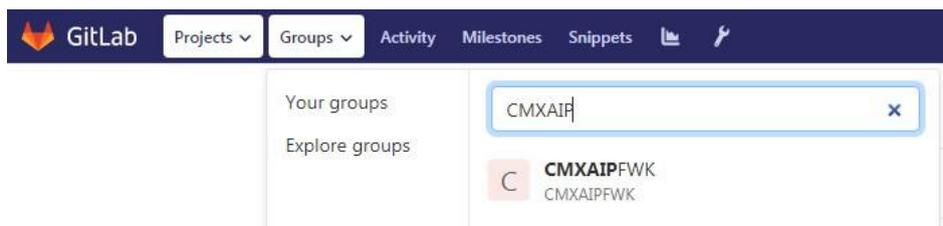
Для успешной работы процессов CI&CD требуется выполнить настройку переменных среды окружения CI&CD в созданных ранее группах проектов.

#### 4.4.1 Настройка переменных CI&CD для группы CMXAIPFWK

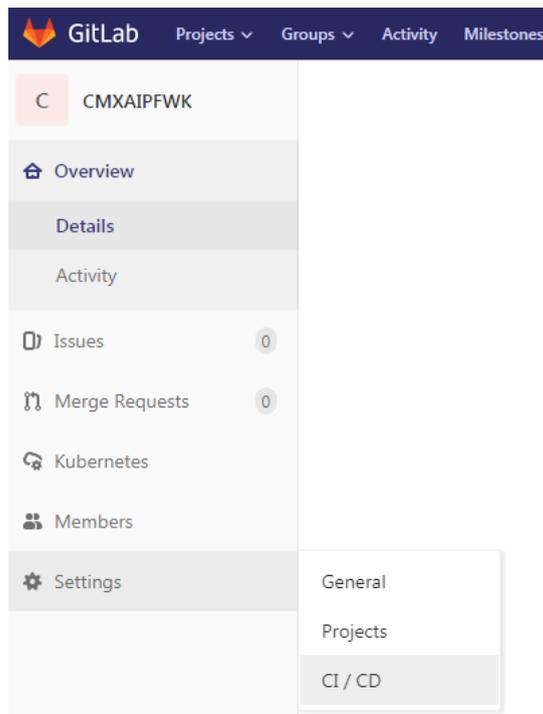
**Шаг 1.** Войти в веб-интерфейс GitLab под своей учетной записью <GitLab USER>

Ссылка для доступа <GitLab URL> (пример: <https://gitlab.infra.k8sdemo.vm.cmx.ru>)

**Шаг 2.** Открыть группу проектов CMXAIPFWK, с помощью поиска в меню Groups

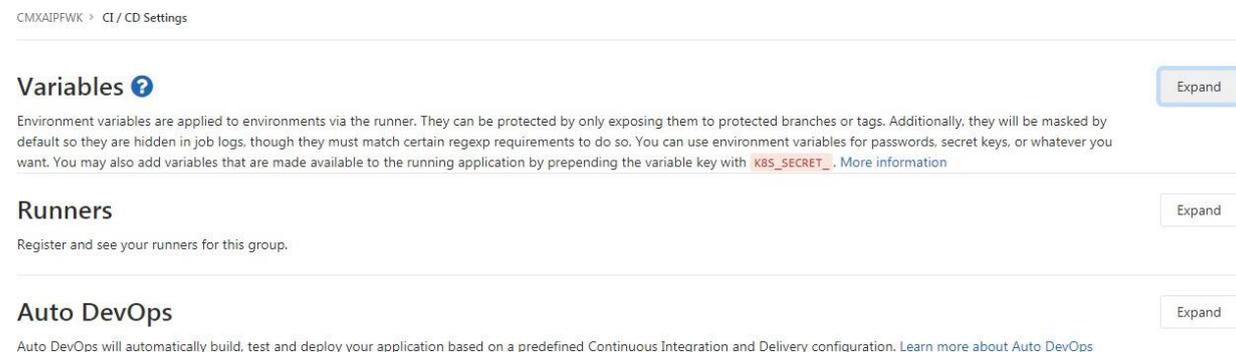


**Шаг 3.** Перейти в настройки CI&CD группы через меню Settings -> CI&CD



Будет осуществлен переход на страницу определения переменных среды окружения.

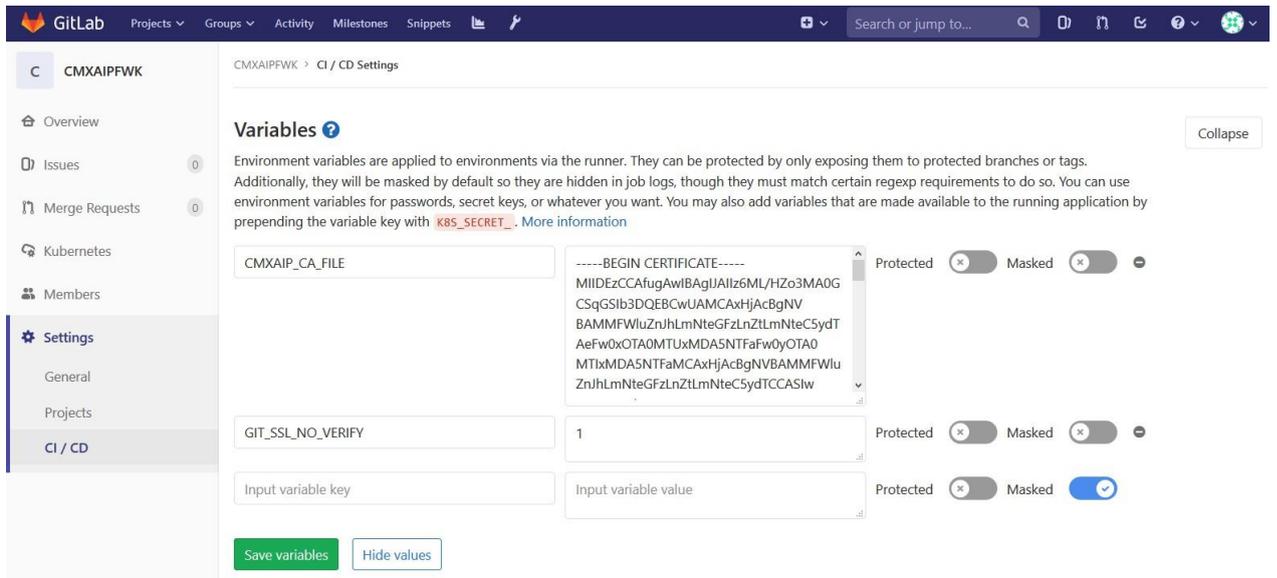
#### Шаг 4. Раскрыть подменю переменных (Variables) с помощью кнопки «Expand»



#### Шаг 5. Создать определение новых переменных среды

**Примечание:** при заполнении значений переменных система выведет предупреждение «Cannot use Masked Variable with current value». Для того, чтобы убрать предупреждение необходимо выключить опцию «Masked» для данной переменной.

- Переменная **CMXAIP\_CA\_FILE**  
В качестве значения указать значение параметра **<Platform CA>**
- Переменная **GIT\_SSL\_NO\_VERIFY**  
В качестве значения указать **1**



**Шаг 6.** Нажать кнопку «Save variables»

#### 4.4.2 Настройка переменных CI&CD для группы интеграционных приложений

Аналогичные пункту 4.4.1 действия требуется повторить для группы интеграционных приложений, созданной в рамках пункта 4.2.2.

### 4.5 СБОРКА И ПУБЛИКАЦИЯ ПРОЕКТОВ

#### 4.5.1 Сборка и публикация проекта msfwk-image-ci

Требуется собрать и опубликовать в GitLab платформы проект msfwk-image-ci из дистрибутива фреймворка.

**Примечание:** Для ОС Windows, здесь и далее по документу подразумевается использование командной строки Git Bash из ПО «Git for Windows» (является пререквизитом для установки фреймворка на рабочую станцию согласно пункту 3.1.2 данного руководства).

**Шаг 1.** Перейти в папку рабочей области

Место размещения папки с рабочей областью определяется в процессе установки дистрибутива фреймворка. См. пункт 5.4 документа «Cinimex Integration Framework. Руководство разработчика.pdf» версии 1.0.0.

**Шаг 2.** Последовательно выполнить следующие инструкции в командной строке

```
cd msfwk-image-ci
git init

git add .
git commit -m "From cinimex-integration-framework-v1.0.0.zip"

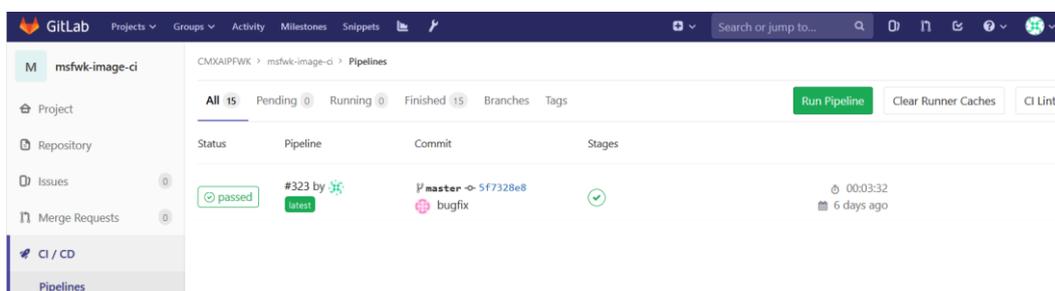
git remote add k8s <GitLab URL>/cmxaipfwk/msfwk-image-ci.git

git push -u k8s master
```

После выполнения последней команды система запросит имя и пароль пользователя в интерактивном режиме. Требуется ввести данные своей учётной записи: <GitLab USER>.

**Шаг 3.** После публикации проекта автоматически запустится процесс CI&CD для сборки образа. Процесс может занять несколько минут.

**Шаг 4.** Проверить результат выполнения CI/CD процесса в проекте CMXAIPFWK/msfwk-image-ci  
Для проверки статуса требуется открыть страницу проекта и перейти в меню CI / CD проекта.  
Статус «passed» сигнализирует об успешном выполнении процесса.



#### 4.5.2 Сборка и публикация проекта msfwk

Для сборки и публикации проекта msfwk требуется выполнить действия, аналогичные пункту 4.4.1

**Шаг 1.** Перейти в папку рабочей области

**Шаг 2.** Последовательно выполнить следующие команды

```
cd msfwk
git init

git add .
git commit -m "From cinimex-integration-framework.v1.0.0.zip"

git remote add k8s <GitLab URL>/cmxaipfwk/msfwk.git

git push -u k8s master
```

**Шаг 3.** Проверить результат выполнения CI/CD процесса в проекте CMXAIPFWK/msfwk.

#### 4.5.3 Сборка проекта msfwk-server-config

**Шаг 1.** Перейти в папку рабочей области

**Шаг 2.** Последовательно выполнить следующие команды

```
cd msfwk-server-config
git init

git add .
git commit -m "From cinimex-integration-framework.v1.0.0.zip"

git remote add k8s <GitLab URL>/cmxaipfwk/msfwk-server-config.git

git push -u k8s master
```

**Шаг 3.** Проверить результат выполнения CI/CD pipeline цикла в проекте CMXAIPFWK/msfwk-server-config.

#### 4.5.4 Сборка и публикация проекта msfwk-server-hawtio

**Шаг 1.** Перейти в папку рабочей области

**Шаг 2.** Последовательно выполнить следующие команды

```
cd msfwk-server-hawtio
git init

git add .
git commit -m "From cinimex-integration-framework.v1.0.0.zip"

git remote add k8s <GitLab URL>/cmxaipfwk/msfwk-server-hawtio.git

git push -u k8s master
```

**Шаг 3.** Проверить результат выполнения CI/CD pipeline цикла в проекте CMXAIPFWK/msfwk-server-hawtio

## 5 СОЗДАНИЕ И ПУБЛИКАЦИЯ ПРОЕКТА РАБОЧЕЙ ОБЛАСТИ (NAMESPACE)

### 5.1 СОЗДАНИЕ ТОКЕНА ДЛЯ ДОСТУПА К РЕЕСТРУ ОБРАЗОВ

Для возможности загрузки образов контейнеров приложений в среду Kubernetes используется специальный secret-объект. Secret-объект создается автоматически в рамках процесса CI&CD по развертыванию проекта рабочей области на основе docker-registry токена.

Docker-registry токен представляет собой файл в формате JSON закодированный в строку base64. Для его создания требуется выполнить следующие действия:

**Шаг 1.** Перейти в папку рабочей области

**Шаг 2.** Закодировать строку с данными авторизации технологической четной записи в строку base64

Для этого в командной строке необходимо выполнить команду

```
echo "msfwk-service-account:<service-account-token>" | base64
```

где `<service-account-token>` - значение обезличенного токена для технологической учетной записи, полученное в результате выполнения шагов инструкции пункта 4.1 настоящего документа.

В результате выполнения команды в консоль будет выведена строка в кодировке base64. Это значение необходимо использовать вместо параметра `<service-account-token-base64>` на следующем шаге.

Пример выполнения команды:

```
[~$ echo "msfwk-service-account:HJXPsjFoPmDYKMNDszR_" | base64
bXNmd2stc2VydmVjZS1hY2NvdW50OkhKWFBzakZvUG1EWUtNTkRze1JfCg==
```

**Шаг 3.** Создать JSON файл с описанием токена

Необходимо создать файл token.json в корне папки рабочей области со следующим содержимым:

```
{
  "auths": {
    "<GitLab DOMAIN>": {
      "auth": "<service-account-token-base64>"
    }
  },
}
```

```
"HttpHeaders": {
  "User-Agent": "Docker-Client/18.06.1-ce (linux)"
}
```

где `<service-account-token-base64>` - строка в кодировке base64, полученная на предыдущем шаге.

Ниже приведен пример заполнения файла:

```
{
  "auths": {
    "registry.infra.k8sdemo.vm.cmx.ru": {
      "auth":
"BXNmd2stc2VydmljZS1hY2NvdW50OkhKWFBzakZvUG1EWUtNTkRzelJfCg=="
    }
  },
  "HttpHeaders": {
    "User-Agent": "Docker-Client/18.06.1-ce (linux)"
  }
}
```

**Шаг 4.** Закодировать файл в строку base64

```
cat token.json | base64 > token.txt
```

**Шаг 5.** Использование токена

Созданный предыдущем шаге файл `token.txt` будет содержать значение, которое необходимо передать в качестве параметра `<Registry TOKEN>` при выполнении команды `generate` из следующего раздела.

## 5.2 СОЗДАНИЕ ПРОЕКТА РАБОЧЕЙ ОБЛАСТИ

### 5.2.1 Сборка maven-archetype проекта

Перед генерацией проекта с рабочей областью требуется собрать проект `msfwk-template-namespace` локально на рабочей станции администратора фреймворка:

**Шаг 1.** Перейти в папку рабочей области

**Шаг 2.** Последовательно выполнить следующие инструкции в командной строке

```
cd msfwk-template-namespace
mvn clean install
```

**Шаг 3.** Проверка результата

В результате выполнения команд в терминале командной должен отобразиться статус «BUILD SUCCESS», сигнализирующий об успешном завершении процесса сборки проекта.

### 5.2.2 Генерация проекта рабочей области

Для генерации проекта рабочей области из шаблона `msfwk-template-namespace` требуется выполнить следующие действия:

**Шаг 1.** Перейти в папку рабочей области



```
git add .
git commit -m "Generated from template"

git remote add k8s <GitLab URL>/msfwk-examples/msfwk-examples-namespace.git
git push -u k8s master
```

### Шаг 3. Проверка результата развертывания проекта

После выкладывания проекта в репозиторий автоматически запустится CI/CD процесс сборки и развертывания namespace на платформе Cinimex Microservice Platform.

Проверить результат выполнения CI/CD процесса можно в меню CI&CD опубликованного проекта MSFWK-EXAMPLES/msfwk-examples-namespace.

Статус «passed» сигнализирует об успешном выполнении процесса.

The screenshot shows the GitLab interface for the project 'examples-namespace'. The left sidebar contains navigation options: Project, Repository, Issues (0), Merge Requests (0), CI / CD, Pipelines, Jobs, Schedules, and Charts. The main content area displays the 'Pipelines' section for 'MSFWK-EXAMPLES > examples-namespace'. It shows a summary of pipeline statuses: All 1, Pending 0, Running 0, and Finished 1. A 'Run Pipeline' button is visible. Below this is a table of pipeline runs:

Status	Pipeline	Triggerer	Commit	Stages	Duration
passed	#40 latest		master -> 3edc20df Generated from template		00:00:21 3 minutes ago