

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ
МОДУЛЯ METASFERA XD: METASFERA COSTCONTROL,
ПРЕДУСТАНОВЛЕННОГО НА ВИРТУАЛЬНУЮ МАШИНУ**

**г. Москва
2025 г.**

ГЛОССАРИЙ

Обозначение	Расшифровка	Примечание
ОС	Операционная система	
ИС	Информационная система	
vCPU	Виртуальный процессор, содержащий одно ядро и занимающий один сокет	
ОП	Операционная система	
Web-интерфейс	Совокупность веб-страниц, предоставляющая пользовательский интерфейс для взаимодействия с сервисом или устройством посредством протокола HTTP и веб-браузера	
Orion ZVirt	Программный продукт виртуализации для операционных систем	

СОДЕРЖАНИЕ

1. УСТАНОВКА METASFERA COSTCONTROL	4
2. УСТАНОВКА Orion ZVIRT	4
3. ИМПОРТИРОВАНИЕ ПОДГОТОВЛЕННОЙ ВИРТУАЛЬНОЙ МАШИНЫ В СРЕДУ ВИРТУАЛИЗАЦИИ ORION ZVIRT	4
3.1 Импорт виртуальной машины.....	4
4. ПЕРВОЕ ВКЛЮЧЕНИЕ И ВХОД В ОПЕРАЦИОННУЮ СИСТЕМУ	4
4.1 Аутентификация в операционной системе	5
5. ЗАПУСК БРАУЗЕРА	5
6. НАСТРОЙКА METASFERA COSTCONTROL	6
6.1 Настройка инфраструктуры.....	7
6.2 Настройка веб-части	8
6.3 Установка модуля.....	10
6.4 Запуск, остановка веб-части	10
6.5 Настройки серверной части.....	10
6.6 Запуск, остановка серверной части.....	15
7. Требования и рекомендации по безопасности	16

1. УСТАНОВКА METASFERA COSTCONTROL

Установка Metasfera CostControl состоит из следующих шагов:

1. Установка программного продукта виртуализации для операционных систем **Orion ZVirt**
2. Импорт образа виртуальной машины с предустановленным модулем

2. УСТАНОВКА Orion ZVIRT

Для инсталляции **Orion ZVirt** необходимо:

- Зайти на официальный сайт <https://www.orionsoft.ru/zvirt/>;
- Скачать установщик, подходящий под вашу операционную систему.
- Произвести инсталляцию программного продукта **Orion ZVirt** в соответствии с инструкцией.

3. ИМПОРТИРОВАНИЕ ПОДГОТОВЛЕННОЙ ВИРТУАЛЬНОЙ МАШИНЫ В СРЕДУ ВИРТУАЛИЗАЦИИ ORION ZVIRT

3.1 Импорт виртуальной машины

Образ виртуальной машины необходимо взять по ссылке, ссылка на архив и пароль архива запрашивается отдельно.

Далее, при необходимости, можно изменить некоторые параметры – например, сменить имя виртуальной машины, уменьшить или увеличить оперативную память или количество ядер процессора, выделяемые для виртуальной машины.

Рекомендуемые требования к виртуальной машине:

1. Не менее 16 гб оперативной памяти;
2. Не менее 50 гб раздела жесткого диска;
3. Не менее 4 vCPU

Минимальные требования к системе:

1. 4 ядра
2. Из расчета 25 пользователей на 1 ядро для расширения
3. 4 ГБ доступной памяти на 1 ядро системы

Поддерживаемые ОС: *nix, Windows (Win2012 R2 и более поздние), в том числе РЕД ОС, Astra Linux, ОС РОСА

Поддерживаемые веб-браузеры: Mozilla Firefox, Microsoft Edge, Apple Safari, Google Chrome, Яндекс Браузер

4. ПЕРВОЕ ВКЛЮЧЕНИЕ И ВХОД В ОПЕРАЦИОННУЮ СИСТЕМУ

4.1 Аутентификация в операционной системе

После того, как операционная система запустится, необходимо выбрать пользователя user и ввести пароль. (См. Рисунок 1)

Пользователь: user (запрашивается отдельно)

Пароль:113311



Рисунок 1 Вход в операционную систему

5. ЗАПУСК БРАУЗЕРА



Рисунок 2 Desktop виртуальной машины

Необходимо нажать на иконку браузера «Яндекс Браузер» (в данной операционной системе установлен по умолчанию), после чего запустится данный браузер. (См. Рисунок 2).
По умолчанию открывается окно с логином в систему. (См. Рисунок 3)
Если оно не открылось, выбираем из закладки.



Рисунок 3 Окно с логином в систему

На странице логина необходимо ввести логин и пароль от системы.

После ввода логина и пароля происходит аутентификация пользователя и его авторизация. Если у пользователя есть доступ к функционалу, то появится стартовая страница.

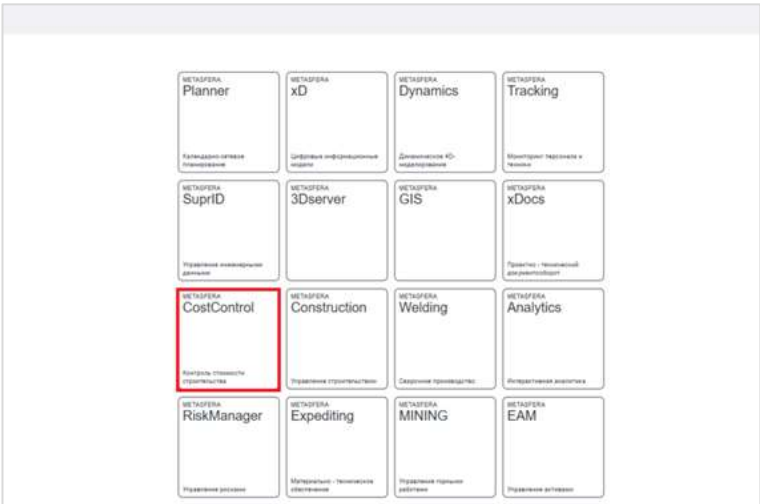


Рисунок 4. Стартовая страница

6. НАСТРОЙКА METASFERA COSTCONTROL

6.1 Настройка инфраструктуры

Для работы системы потребуются сервисы инфраструктуры. На виртуальной машине предустановлены:

- **СУБД PostgreSQL 15.5**
- **Nodejs 18.0**
- **Dotnet 6.0**
- **pm2 5.2.2**
- **RabbitMQ 3.9**
- **Redis 7.0**
- **Nginx 1.23**

Установка СУБД PostgreSQL

Установить и запустить СУБД PostgreSQL 15 на сервере СУБД:

```
dnf install -y postgresql15-server postgresql15-contrib
```

```
postgresql-15-setup initdb
```

```
systemctl enable postgresql-15 --now
```

```
systemctl status postgresql-15
```

Отредактируем:

```
/var/lib/pgsql/15/data/postgresql.conf
```

```
установить listen_addresses = '*'
```

Установка nodejs, npm, dotnet

Выполнить на сервер_приложений:

```
dnf install -y nodejs npm
```

```
dnf install dotnet-sdk-6.0.x86_64 -y
```

Установка pm2

```
npm install pm2 -g
```

Установка RabbitMQ

Выполнить на сервер_приложений:

```
dnf -y install rabbitmq-server
```

```
systemctl enable rabbitmq-server --now
```

```
systemctl status rabbitmq-server
```

```
rabbitmqctl add_user rmq_user (указать пароль rmq_user)
```

```
rabbitmqctl set_user_tags rmq_user administrator
```

```
rabbitmqctl set_permissions -p / rmq_user ".*" ".*" ".*"
```

Установка Redis

Выполнить на сервер_приложений:

Установить Redis:

```
dnf install -y redis
```

Отредактировать /etc/redis/redis.conf

```
bind 0.0.0.0
protected-mode no
Добавить сервис в автозапуск и запустить:
systemctl enable redis --now
systemctl status redis
```

Установка nginx

Выполнить на веб-сервере:

Отключение SELINUX

```
setenforce 0
```

```
sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/syscon-
fig/selinux
```

```
dnf -y install nginx
```

```
systemctl enable nginx --now
```

```
systemctl start nginx
```

```
systemctl status nginx
```

6.2 Настройка веб-части

К конфигурации web-интерфейса системы относятся параметры, необходимые для обновления версии и отображения системы.

Настройка порта соединения, корневой директории сайта и способа отображения находятся в конфигурационном файле веб-сервера.

Конфигурационный файл находится:

```
/etc/nginx/nginx.conf
```

И выглядит следующим образом:

```
user nginx;
```

```
worker_processes auto;
```

```
error_log /var/log/nginx/error.log notice;
```

```
pid /run/nginx.pid;
```

```
# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
```

```
include /usr/share/nginx/modules/*.conf;
```

```
events {
```

```
    worker_connections 1024;
```

```
}
```

```
http {
```

```
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
```

```
                    '$status $body_bytes_sent "$http_referer" '
```

```
                    '"$http_user_agent" "$http_x_forwarded_for"';
```

```
    access_log /var/log/nginx/access.log main;
```

```
    sendfile        on;
```

```
    tcp_nopush      on;
```

```
keepalive_timeout 65;
types_hash_max_size 4096;
include          /etc/nginx/mime.types;
default_type     application/octet-stream;

# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/nginx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;

server {
    listen      80;
    listen      [::]:80;
    server_name _;
    root        /usr/share/nginx/html;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;
    error_page 404 /404.html;
    location = /404.html {
    }
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

Конфигурация frontend приложений.

Frontend приложения построены на подходе «Module Federation», что позволяет добиться гибкости и масштабируемости приложений, создавать более сложные и функциональные приложения.

В состав Frontend входят следующие приложения:

1. admin (модуль администрирования)
2. host (модуль системных задач)
3. auth (модуль авторизации)
4. systask (модуль асинхронных задач)
5. wrapper (модуль внешней интеграции)
6. objcon (модуль конструктора объектов)
7. pdf-viewer (модуль просмотра pdf файлов)

Конфигурация модулей идентична. Рассмотрим на примере конфигурации сервиса “HOST”. Ниже приведем отдельные его части с пояснением того, что они значат.

1. server {
listen 7000; - указывает, какой порт готов принимать входящее соединения

2. `server_name localhost;` - указывает адрес, на котором работает веб сервер
3. `root /etc/nginx/www/host;` - указывает корневую директорию.
4. `location / {` - инструкция, указывающая на корневой каталог с ответом на ошибочные запросы
`try_files $uri $uri/ /index.html =404; }`
5. `location /api/ {` - блок инструкций по указанию работы с серверной частью части веб интерфейса
6. `proxy_pass http://localhost:4300;` - адрес серверной части
7. `proxy_http_version 1.1;` - версия запросов, используемая в ПО
8. `proxy_set_header Upgrade $http_upgrade;` - системные настройки части переадресации запросов
9. `proxy_set_header Connection 'upgrade';` - системные настройки части переадресации запросов
10. `proxy_set_header Host $host;` - системные настройки части переадресации запросов
11. `proxy_cache_bypass $http_upgrade;` - системные настройки части переадресации запросов
12. `client_max_body_size 100M;` - максимальный допустимый размер файлов для передачи через веб-интерфейс
13. `proxy_read_timeout 300;` - таймаут ожидания, после которого веб-приложение выдаст ошибку
14. `proxy_connect_timeout 300;` - таймаут ожидания, после которого веб-приложение выдаст ошибку в варианте 2
15. `proxy_send_timeout 300;` - таймаут ожидания, после которого веб-приложение выдаст ошибку в варианте 3
16. `send_timeout 300;` - таймаут ожидания, после которого веб-приложение выдаст ошибку в варианте 4

6.3 Установка модуля

Распаковать архив `CostControl.zip` в каталог `/tmp/CostControl`.

Для установки модуля необходимо выполнить скрипт `run.sql` из каталога `/tmp/CostControl`

6.4 Запуск, остановка веб-части

Для запуска веб-интерфейса необходимо выполнить команду
`systemctl start nginx`

Для остановки веб-интерфейса необходимо выполнить команду
`systemctl stop nginx`

6.5 Настройки серверной части

Серверная часть приложения состоит из следующих сервисов:

1. API

2. SECURITY
3. INTEGRATION
4. DOCUMENTS
5. LOGGER
6. FREPORT
7. FILEMANAGER
8. OPERATOR

Описание конфигурации API

Файлы конфигурации API располагаются по следующему пути
/opt/CostControl/monolith/config/

httpPort: 4300 – порт приложения для HTTP запросов

httpsPort: 4301 – порт приложения для HTTPS запросов

NODE_ENV_: PROD - обозначение среды, на которой запущен сервер

secret: Qbw1GG - задание пароля

token_expires: 7d - время жизни ключа авторизации

token_expiresComment: Время жизни токена

time_to_check_expire: 1440 Время в секундах жизни токена до его проверки

smtp: - параметры для подключения уведомлений SMTP

email: name

user: user

password: pass

host: mail.cs.ru

port: 25

tls: true

timeout: 60000

reporting:

defaultReportPath: /srv/uploads – директория для выгрузки отчетов

- type: postgres – тип БД

database: "testdb" – название БД

host: "localhost" – сервер расположения БД

username: user – пользователь для подключения к БД

password: pass – пароль для подключения к БД

port: 5432 – порт для подключения к БД

connectionTimeout: 60000 – таймаут подключения к БД

securityService: - параметры подключения к сервисам, использующих rabbitmq

transport: 5

options:

urls: [

'=> `amqp://\${user}:\${password}@localhost:5672`' – Указание пользователя, пароля и адреса rabbitmq

```
]
queue: security_queue
prefetchCount: 1
queueOptions:
  durable: true
```

параметры подключения к сервису FREPORT

```
fastReportService:
  host: 'http://localhost'
  port: 9876
```

параметры подключения к сервису DOCUMENTS

```
documentRenderServiceHttp:
  host: 'http://localhost'
  port: 3000
```

параметры подключения к сервису FILEMANAGER

```
fileServiceHttp:
  host: 'http://localhost'
  port: 7070
```

uploadPath: /srv/uploads/ - путь сохранения импортируемых файлов

fileSize: 524288000 - Максимальный размер загружаемого файла

Описание конфигурации SECURITY

Файл конфигурации SECURITY располагается по следующему пути
/opt/CostControl/safety/config

secret: Qbw1GG - задание пароля

token_expires: 7d - Время жизни токена

time_to_check_expire: 1440 - Время в секундах жизни токена до его проверки

time_to_check_expireComment: Время в секундах жизни токена до его проверки

host: localhost - адрес сервиса

port: 3002 - порт, используемый сервисом.

isLdap: false – Проверяет, является ли потенциально авторизованный пользователь участником LDAP

isSSO: false - Включить/Отключить возможность авторизации через SSO

isLimitingLoginAttempts: false Включить/Отключить возможность ограничивать пользователя на попытка ввода пароля в систему

isDecryptPassword: false - Включить/Отключить шифрования пароля

ldap: - настройка параметров подключения к серверу LDAP

url: ldap://cs.local:389

baseDN: DC=csDC=local

rabbitmq: { - настройка параметров подключения к rabbitmq
user: user - пользователь для подключения к rabbitmq
password: pass - пароль для подключения к rabbitmq
host: localhost - хост на котором запущен rabbitmq
port: 5672 - порт на котором запущен rabbitmq
queue: security_queue - название очереди для сервиса

- type: postgres – тип БД
database: testdb – название БД
host: localhost – сервер расположения БД
user: user – пользователь для подключения к БД
password: pass – пароль для подключения к БД
port: 5432 – порт для подключения к БД
connectionTimeout: 60000 – таймаут подключения к БД

Описание конфигурации INTEGRATION

Файл конфигурации INTEGRATION располагается по следующему пути
/opt/CostControl/integration/config/

параметры подключения к сервису API
monolith:

type: monolith
apiurl: 'http://localhost:4300'

параметры подключения к сервису Freport

fastReportService:
host: 'http://localhost'
port: 9876
defaultReportPath: /srv/uploads

параметры подключения к сервису FILEMANAGER

fileServiceHttp:
host: 'http://localhost'
port: 7070

apiBaseUrl: http://localhost:3000 - адрес для подключения сервиса аналитики
apiStartProcess: "/etl/schemata/:schemaId/pipelines" – параметры подключения
apiBaseUrl

user: user - пользователь для подключения к rabbitmq
password: pass - пароль для подключения к rabbitmq
host: localhost - хост на котором запущен rabbitmq
port: 5672 - порт на котором запущен rabbitmq
outputQueues: - Параметры для подключения к очередям rabbitmq
eventQueue:
queue: event_queue

host: 'localhost' – хост подключения к redis
port: 6379 - хост на котором запущен redis
ksp_task_ttl: 28800 - Время жизни кэш для экрана работ

Описание конфигурации DOCUMENTS

Файл конфигурации DOCUMENTS-MICROSERVICE располагается по следующему пути /opt/CostControl/documents-service/config/
rabbitmq: { - настройка параметров подключения к rabbitmq
user: user - пользователь для подключения к rabbitmq
password: pass - пароль для подключения к rabbitmq
host: localhost - хост на котором запущен rabbitmq
port: 5672 - порт на котором запущен rabbitmq
queue: document_render_queue - название очереди для сервиса
параметры подключения к сервису FILEMANAGER
fileServiceHttp:
host: http://localhost:7070

Описание конфигурации LOGGER

Файл конфигурации LOGGER располагается по следующему пути /opt/CostControl/logger/config/
type: postgres – тип БД
database: pmdb_log – название БД
server: localhost – сервер расположения БД
user: user – пользователь для подключения к БД
password: pass – пароль для подключения к БД
port: 5432 – порт для подключения к БД

rabbitmq: { - настройка параметров подключения к rabbitmq
user: user - пользователь для подключения к rabbitmq
password: pass - пароль для подключения к rabbitmq
hostname: localhost - хост на котором запущен rabbitmq
port: 5672 - порт на котором запущен rabbitmq

Описание конфигурации FREPORT

Файл конфигурации FREPORT располагается по следующему пути /opt/CostControl/fr/appsettings.json
DbAliasList - набор БД для подключения может иметь несколько источников
Name: DbPg - название источника для подключения
DataSource: localhost:5432 – сервер расположения БД с указанием порта
InitialCatalog: testdb – название БД
UserID: user – пользователь для подключения к БД
Password: pass – пароль для подключения к БД

Type: Pgsq1 – тип БД

Описание конфигурации FILEMANAGER

Файл конфигурации FILEMANAGER располагается по следующему пути
/opt/CostControl/filemanager/config

pg_host: 'localhost' – сервер расположения БД
pg_port: 5432 – порт для подключения к БД
pg_database: 'testdb' – название БД
pg_username: 'user' – пользователь для подключения к БД
pg_password: 'pass' – пароль для подключения к БД

Описание конфигурации OPERATOR

Файл конфигурации OPERATOR располагается по следующему пути
/opt/CostControl/operator/config/

dbId: CostControl – тип БД
database: "testdb" – название БД
host: "localhost" – сервер расположения БД
username: user – пользователь для подключения к БД
password: pass – пароль для подключения к БД
port: 5432 – порт для подключения к БД

rabbitmq: { - настройка параметров подключения к rabbitmq
user: user - пользователь для подключения к rabbitmq
password: pass - пароль для подключения к rabbitmq
hostname: localhost - хост на котором запущен rabbitmq
port: 5672 - порт на котором запущен rabbitmq

jobscron

cron-send-event-weekly-at-mdnt: – название запланированной задачи
executorName: executor-send-event-weekly-at-mdnt – название исполняемого файла запланированной задачи
enable: true – параметр активности запланированной задачи
pattern: '0 0 0 * * ' - расписание выполнения запланированной задачи

6.6 Запуск, остановка серверной части

Для того, чтобы запустить серверную часть, используется PM2 - диспетчер процессов для среды выполнения Node.js

1. cd /opt/CostControl/
2. pm2 start pm2.config.js
3. Просмотр запущенных модулей pm2 ls

Для того, чтобы остановить серверную часть, необходимо выполнить две команды в терминале

4. cd /opt/CostControl/
5. pm2 stop all

7. Требования и рекомендации по безопасности

- Пароли, содержащиеся в конфигурационных файлах, зашифрованы с использованием криптостойкого алгоритма шифрования (AES256).
- Доступ к стендам системы должен быть ограничен и предоставляться по согласованию с владельцем системы.
- Запрещается передавать учетные данные для доступа к стендам системы и входа в систему третьим лицам.
- Подключение к веб-интерфейсу системы должно осуществляться с использованием TLS.
- К работе с системой должен допускаться только персонал, прошедший обучение.
- Требования к защите периметра безопасности инфраструктуры владельца системы в данном документе не рассматриваются, т.к. выходят за рамки ответственности системы.