

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

**г. Москва  
2023 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

<b>1. УСТАНОВКА METASFERA PLANNER</b> .....	4
<b>2. УСТАНОВКА Orion ZVIRT</b> .....	4
<b>3. ИМПОРТИРОВАНИЕ ПОДГОТОВЛЕННОЙ ВИРТУАЛЬНОЙ МАШИНЫ В СРЕДУ ВИРТУАЛИЗАЦИИ ORION ZVIRT</b> .....	4
<b>3.1 Импорт виртуальной машины</b> .....	4
<b>4. ПЕРВОЕ ВКЛЮЧЕНИЕ И ВХОД В ОПЕРАЦИОННУЮ СИСТЕМУ</b> .....	5
<b>4.1 Аутентификация в операционной системе</b> .....	5
<b>5. ЗАПУСК БРАУЗЕРА</b> .....	6
<b>6. НАСТРОЙКА METASFERA PLANNER</b> .....	7
<b>6.1 Настройка веб-части</b> .....	7
<b>6.2 Запуск/остановка веб-части</b> .....	9
<b>6.3 Настройки серверной части</b> .....	9
<b>6.4 Запуск\остановка серверной части</b> .....	14
<b>6.5 Запуск\остановка системы управления базой данных (СУБД)</b> .....	15
<b>6.6 Конфигурация СУБД PostgreSQL</b> .....	15

## 1. УСТАНОВКА METASFERA PLANNER

Установка Metasfera Planner 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:

1. Red Hat Enterprise Linux (RHEL) (32-bit or 64-bit)
2. Novel SUSE Linux Enterprise Server (SLES) (32-bit or 64-bit)
3. Microsoft Windows (32-bit or 64-bit)
4. Apple Mac OS
5. Solaris SPARC

6. Fedora
7. Debian Linux
8. HP-UX
9. FreeBSD
10. CentOS
11. Ubuntu

Поддерживаемые веб-браузеры:

1. Mozilla Firefox
2. Microsoft Internet Explorer
3. Apple Safari
4. Google Chrome
5. 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.

## 6. НАСТРОЙКА METASFERA PLANNER

### 6.1 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. cnp (модуль календарно-сетевого планирования)
7. objcon (модуль конструктора объектов)

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

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

## 6.2 Запуск/остановка веб-части

Для запуска веб-интерфейса необходимо выполнить команду  
systemctl start nginx

Для остановки веб-интерфейса необходимо выполнить команду  
systemctl stop nginx

## 6.3 Настройки серверной части

Серверная часть приложения состоит из следующих сервисов:

1. API
2. SECURITY
3. INTEGRATION
4. ADMINSERVICE
5. KSP-MICROSERVICE
6. DOCUMENTS-MICROSERVICE
7. MPP-MICROSERVICE
8. CALC-SCHEDULER

### Описание конфигурации API

Файлы конфигурации API располагается по следующему пути  
/opt/planner/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

import\_ksg:

importDirectory: /srv/import-ksg – директория выгрузки ksg

url: <http://localhost:8078/api/import/mmp> - адрес сервиса выгрузки ksg

chunkSizeMb: 64 – максимальный размер файла

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

calcschedule\_http:

apiurl: 'http://localhost:8002' – адрес для подключения сервиса calcschedule

---

- type: postgres – тип БД

database: "testdb" – 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

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

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

fileSize: 524288000 - Максимальный размер загружаемого файла

## Описание конфигурации SECURITY

Файл конфигурации SECURITY располагается по следующему пути  
/opt/planner/safety/config.json  
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 пользова-  
тель 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  
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  
connectionTimeout: 60000 – таймаут подключения к БД

### Описание конфигурации INTEGRATION

Файл конфигурации INTEGRATION располагается по следующему пути  
/opt/planner/integration/config/  
import\_ksg:  
importDirectory: /srv/import-ksg – директория выгрузки ksg  
url: <http://localhost:8078/api/import/mmp> - адрес сервиса выгрузки ksg  
chunkSizeMb: 64 – максимальный размер файла

- type: postgres – тип БД  
database: "testdb" – 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  
connectionTimeout: 60000 – таймаут подключения к БД

calcschedule-http:

type: calcschedule-http

apiurl: 'http://localhost:8002' - адрес для подключения сервиса calcschedule

monolith:

type: monolith

apiurl: 'http://localhost:4300' - адрес для подключения сервиса monolith

apiBaseUrl: <http://10.0.1.112:3000> - 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 ADMINSERVICE**

Файл конфигурации ADMINSERVICE располагается по следующему пути /opt/planner/adminservice/config

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: admin\_new\_queue - название очереди для сервиса

- type: postgres – тип БД  
database: testdb – 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  
connectionTimeout: 60000 – таймаут подключения к БД

#### Описание конфигурации **KSP-MICROSERVICE**

Файл конфигурации KSP-MICROSERVICE располагается по следующему пути /opt/planner/ksp-microservice/config/

- type: postgres – тип БД  
database: testdb – 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  
connectionTimeout: 60000 – таймаут подключения к БД

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: ksp\_queue - название очереди для сервиса

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-MICROSERVICE**

Файл конфигурации DOCUMENTS-MICROSERVICE располагается по следующему пути /opt/planner/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

### Описание конфигурации MPP-MICROSERVICE

Файл конфигурации MPP-MICROSERVICE располагается по следующему пути  
/opt/planner/mpp-microservice/appsettings.json

```
App: {  
  DeleteMPP: false - параметр удаления mpp после импорта  
}  
DB: {  
  Driver: PostgreSQL - тип драйвера  
  DataSource: localhost - хост БД  
  Port: 5432 - порт БД  
  UserID: padmin - пользователь БД  
  Password: pass - пароль к БД  
  InitialCatalog: testdb - имя БД  
  ApplicationName: PLANNER  
  CommandTimeout: 60000 - время таймаута
```

### Описание конфигурации CALC-SCHEDULER

Файл конфигурации CALC-SCHEDULER располагается по следующему пути  
/srv/config/. Сервис запущен как служба с автозапуском.

Проверка состояния сервиса `systemctl status back-service.service`

```
- type: postgres - тип БД  
database: "testdb" - имя БД  
server: "localhost" - хост БД  
user: "padmin" - пользователь БД  
password: "password" - пароль к БД  
port: 10265 - порт БД  
connectionTimeout: 60000 - время таймаута
```

## 6.4 Запуск\остановка серверной части

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

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

id	name	namespace	version	mode	pid	uptime	↓	status
3	adminservice	default	N/A	fork	4695	10s	0	online
0	api	default	N/A	fork	4676	10s	0	online
5	document-service	default	N/A	fork	4710	10s	0	online
2	integration	default	N/A	fork	4683	10s	0	online
4	ksp-service	default	N/A	fork	4783	10s	0	online
6	app	default	N/A	fork	4711	10s	0	online
1	security-microservice	default	N/A	fork	4677	10s	0	online

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

4. `cd /opt/planner/`
5. `pm2 stop all`

## 6.5 Запуск\остановка системы управления базой данных (СУБД)

Для того, чтобы запустить СУБД, необходимо выполнить команду `systemctl start postgresql`

Для того, чтобы остановить СУБД, необходимо выполнить команду `systemctl stop postgresql`

## 6.6 Конфигурация СУБД PostgreSQL

Основной конфигурационный файл — `postgresql.conf`

Если необходимо, при запуске сервера, значения параметров можно передать команде `postgres` в аргументе командной строки `-c`.

Например:

```
postgres -c log_connections=yes -c log_destination='syslog'
```