

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ
МОДУЛЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
METASFERA XD: METASFERA DYNAMICS**

г. Москва
2026 г.

Оглавление

1.	Введение	3
2.	Отключение SELinux, обновление ОС, перезагрузка.....	3
3.	Установка и настройка СУБД PostgreSQL	3
4.	Установка и настройка nodejs, npm, pm2	4
5.	Установка и настройка RabbitMQ	4
6.	Установка и настройка Redis	4
7.	Установка и настройка Nginx	5
8.	Установка и настройка Tomcat	5
9.	Установка и настройка GeoServer	5
10.	Установка и настройка MinIO.....	6
10.1.	Загрузка данных MinIO	6
11.	Установка и настройка модуля Metasfera xD: Metasfera Dynamics	7
11.1.	Развертывание баз данных PostgreSQL	7
11.2.	Установка и настройка модуля Image server.....	8
11.3.	Установка и настройка модуля GIS	10
11.4.	Установка и настройка модулей BackOffice.....	13
11.5.	Установка и настройка модуля Etl.....	19
11.6.	Установка и настройка микросервисов	22
11.7.	Конфигурация веб-сервера.....	25
11.7.1.	Установка и настройка модуля Metasfera xD.GIS	26
11.7.2.	Установка и настройка Shell.....	29
11.7.3.	Установка и настройка модуля Metasfera xD: Metasfera Dynamics	31
11.7.4.	Выполнение перезапуска Nginx.....	32
12.	Запуск браузера	32

1. Введение

Настоящее руководство описывает процесс установки и настройки программного обеспечения Руководство по установке и использованию модуля программного обеспечения Metasfera xD: Metasfera Dynamics.

Для успешной установки предполагается, что дистрибутив уже размещён в директории /srv/distr, а компонент rpm-api — установлен и настроен. Далее описывается состав Metasfera Dynamics (Таблица 1).

Таблица 1. Компоненты Metasfera Dynamics

Компонент	Описание
backofficexd	Сервис геометрических 3D вычислений
etl	Сервис аналитических вычислений
gis	Основной API Metasfera xD
image_server	Сервис для работы с изображениями
microservices	Микросервисы
sites	Web-компоненты для всех сервисов

Далее будет описан процесс установки и настройки модуля программного обеспечения Metasfera xD: Metasfera Dynamics для дистрибутива RedOS 8.

2. Отключение SELinux, обновление ОС, перезагрузка

```
sed -i "s/SELINUX=enforcing/SELINUX=permissive/"  
/etc/selinux/config  
setenforce 0  
dnf update -y  
reboot
```

3. Установка и настройка СУБД PostgreSQL

Установить и запустить СУБД PostgreSQL 15:

```
dnf install --setopt=install_weak_deps=False postgresql15-  
server postgresql15-contrib postgis-pgsql15 pgagent_15 -y  
postgresql-15-setup initdb  
systemctl enable postgresql-15 --now
```

```
systemctl status postgresql-15
```

Отредактировать файл /var/lib/pgsql/15/data/pg_hba.conf, добавить в ipv4 строку:

```
host all all 0.0.0.0/0 scram-sha-256
```

```
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 scram-sha-256
host all all 0.0.0.0/0 scram-sha-256
# IPv6 local connections:
host all all ::1/128 scram-sha-256
# Allow replication connections from localhost, by a user with the
```

Перезапустить PostgreSQL:

```
systemctl restart postgresql-15
```

4. Установка и настройка nodejs, npm, pm2

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

```
npm install pm2 -g
```

```
pm2 startup
```

```
systemctl start pm2-root
```

```
systemctl status pm2-root
```

5. Установка и настройка RabbitMQ

Во время создания пользователя нужно будет указать пароль:

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

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

```
systemctl status rabbitmq-server
```

```
rabbitmq-plugins enable rabbitmq_management
```

```
rabbitmqctl add_user admin
```

```
rabbitmqctl set_user_tags admin administrator
```

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

6. Установка и настройка Redis

Установить Redis:

```
dnf install redis -y
```

Настроить `/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
```

7. Установка и настройка Nginx

```
dnf install nginx -y
systemctl enable nginx --now
systemctl status nginx
```

8. Установка и настройка Tomcat

```
dnf install java-17-openjdk-headless tomcat fontconfig wget
unzip -y
echo CATALINA_OPTS=-
DALLOW_ENV_PARAMETRIZATION=true >>/etc/sysconfig/tomcat
systemctl enable tomcat --now
systemctl status tomcat
```

9. Установка и настройка GeoServer

```
cd /tmp
wget
https://sourceforge.net/projects/geoserver/files/GeoServer/
2.22.1/geoserver-2.22.1-war.zip
unzip geoserver-2.22.1-war.zip
unzip geoserver.war -d /var/lib/tomcat/webapps/geoserver
rm -rf /var/lib/tomcat/webapps/geoserver/data/*
chown -R tomcat:tomcat /var/lib/tomcat/webapps/geoserver
systemctl restart tomcat
systemctl status tomcat
```

Создать файл настроек `/var/lib/tomcat/webapps/geoserver/data/geoserver-environment.properties`:

```
cat
<<EOF >/var/lib/tomcat/webapps/geoserver/data/geoserver-
environment.properties
POSTGIS_HOST=127.0.0.1
POSTGIS_USER=osm
POSTGIS_PORT=5432
POSTGIS_PASS=<pg_pass>
POSTGIS_DB_CGP=cgp
EOF
```

Проверка geoserver: в браузере открыть url: http://<server_ip>:8080/geoserver, по умолчанию пользователь admin, пароль geoserver. Изменить по необходимости.

10. Установка и настройка MinIO

Установить MinIO:

```
dnf install minio minio-client -y
ln -s /usr/bin/minio-client /usr/local/bin/mcli
```

Произвести настройки в файле */etc/minio/minio.conf*:

```
cat <<EOF >/etc/minio/minio.conf
MINIO_VOLUMES="/srv/minio/data/"
MINIO_ROOT_USER=<miniouser>
MINIO_ROOT_PASSWORD=<miniopass>
EOF
```

Включить и запустить MinIO:

```
systemctl enable minio --now
systemctl status minio
```

10.1. Загрузка данных MinIO

Для удобства работы, можно настроить алиас:

```
mcli alias set local http://localhost:9000 <miniouser>
<miniopass>
```

Создать бакет 3d-assets:

```
mcli mb local/3d-assets
```

Загрузить данные в бакет:

```
mcli cp -r /srv/distr/metasfera/libs/ local/3d-  
assets/models_3d/libs/
```

Разрешить публичный доступ к данным:

```
mcli anonymous set download local/3d-assets
```

11. Установка и настройка модуля Metasfera xD: Metasfera Dynamics

11.1. Развертывание баз данных PostgreSQL

```
su - postgres  
psql  
CREATE ROLE osm WITH LOGIN SUPERUSER PASSWORD '<pg_pass>';  
CREATE DATABASE bim WITH OWNER osm;  
CREATE DATABASE bim_models WITH OWNER osm;  
CREATE DATABASE cgp WITH OWNER osm;  
CREATE DATABASE etl WITH OWNER osm;  
CREATE DATABASE gis WITH OWNER osm;  
CREATE DATABASE model_3d WITH OWNER osm;  
CREATE DATABASE model_processing WITH OWNER osm;  
exit  
pg_restore -d bim /srv/distr/metasfera/db/bim.dump  
pg_restore -d bim_models  
/srv/distr/metasfera/db/bim_models.dump  
pg_restore -d cgp /srv/distr/metasfera/db/cgp.dump  
pg_restore -d etl /srv/distr/metasfera/db/etl.dump  
pg_restore -d gis /srv/distr/metasfera/db/gis.dump  
pg_restore -d model_3d  
/srv/distr/metasfera/db/model_3d.dump  
pg_restore -d model_processing  
/srv/distr/metasfera/db/model_processing.dump  
exit
```

11.2. Установка и настройка модуля Image server

Создать директорию для хранения изображений:

```
mkdir -p /srv/storage/sessions
```

Распаковать дистрибутив:

```
tar xf /srv/distr/metasfera/image_server.tar.xz -C /opt
```

Создать файл настроек `/opt/image_server/app_settings.json`:

```
cat <<EOF >/opt/image_server/app_settings.json
{
    "auth_settings": {
        "host": "",
        "port": 0,
        "pmc_auth": true,
        "pmc_validate": true,
        "validate_token": false
    },
    "pmc_settings": [
        {
            "host": "<PMC_HOST>",
            "port": <PMC_PORT>,
            "use_project_tree": false
        }
    ],
    "storage_settings": {
        "base_path": "/srv/storage/",
        "sessions_path": "sessions/"
    },
    "gis_server_settings": {
        "host": "127.0.0.1",
        "port": 4500
    },
    "log_settings": {
        "use_stdout": true,
        "product": "image_server"
    },
}
```

```
"unicorn_settings": {
  "host": "0.0.0.0",
  "port": 4502
},
"minio_server_settings": {
  "host": "127.0.0.1",
  "port": 9000,
  "access_key": "<miniouser>",
  "secret_key": "<miniopass>",
  "secure": false,
  "files_batch_upload": 4,
  "default_bucket": "3d-assets",
  "default_bucket_path": "source",
  "default_cloud_point_path": "cloud_points",
  "bucket_media": "storage"
},
"rabbitmq_server_settings": {
  "host": "127.0.0.1",
  "port": 5672,
  "user": "admin",
  "pwd": "<rmq_pass>",
  "exchange": "model_processing",
  "virtual_host": "/"
},
"backoffice_xd_server_settings": {
  "host": "127.0.0.1",
  "port": 4508,
  "web_service_model_processing_host": "127.0.0.1",
  "web_service_model_processing_port": 4509
},
"encrypted_config": false
}
EOF
```

Создать сервис системного демона image_server:

```
cat <<EOF >/etc/systemd/system/image_server.service
```

```
[Unit]
Description=Image server service
After=network.target

[Service]
User=root
WorkingDirectory=/opt/image_server
LimitNOFILE=4096
ExecStart=/opt/image_server/image_server
Restart=on-failure
RestartSec=5s

[Install]
WantedBy=multi-user.target
EOF
```

Включить и запустить системный демон image_server:

```
systemctl enable image_server --now
systemctl status image_server
```

11.3. Установка и настройка модуля GIS

Распаковать дистрибутив:

```
tar xf /srv/distr/metasfera/gis.tar.xz -C /opt
```

Создать файл настроек /opt/gis/app_settings.json:

```
cat <<EOF >/opt/gis/app_settings.json
{
    "db_settings": {
        "host": "127.0.0.1",
        "port": 5432,
        "user": "osm",
        "pwd": "<pg_pass>",
        "db_name_gis": "gis",
        "db_name_3d": "model_3d"
    },
    "auth_settings": {
```

```
        "host": "",
        "port": 0,
        "pmc_auth": true,
        "pmc_validate": true,
        "validate_token": false
    },
    "pmc_settings": [
        {
            "host": "<PMC_HOST>",
            "port": <PMC_PORT>,
            "use_project_tree": false
        }
    ],
    "geoserver_settings": {
        "host": "127.0.0.1",
        "port": 8080,
        "user": "admin",
        "pwd": "geoserver"
    },
    "cgp_settings": {
        "host": "127.0.0.1",
        "port": 5432,
        "user": "osm",
        "pwd": "<pg_pass>",
        "db_name": "cgp"
    },
    "log_settings": {
        "exception_in_resp": true,
        "use_stdout": true,
        "level": "info",
        "product": "gis"
    },
    "unicorn_settings": {
        "host": "0.0.0.0",
        "port": 4500
    }
}
```

```
    },
    "redis_settings": {
        "host": "127.0.0.1",
        "port": 6379,
        "db_index": 0,
        "cache_prefix": "fastapi_cache",
        "cache_namespace": "gis",
        "ttl": 86400,
        "enabled": true,
        "update_time": 14400,
        "resolve_host": "127.0.0.1"
    },
    "rabbitmq_settings": {
        "host": "127.0.0.1",
        "port": 5672,
        "user": "admin",
        "pwd": "<rmq_pass>",
        "callback_queue": "output_data",
        "publish_queue": "input_data",
        "blocked_connection_timeout": 65535,
        "heartbeat": 65535,
        "virtual_host": "/"
    },
    "image_server_settings": {
        "host": "127.0.0.1",
        "port": 4502
    },
    "bim_settings": {
        "host": "127.0.0.1",
        "port": 4508
    },
    "encrypted_config": false
}
EOF
```

Создать сервис системного демона gis:

```
cat <<EOF >/etc/systemd/system/gis.service
[Unit]
Description=GIS service
After=network.target

[Service]
User=root
WorkingDirectory=/opt/gis
LimitNOFILE=4096
ExecStart=/opt/gis/gis
Restart=on-failure
RestartSec=5s

[Install]
WantedBy=multi-user.target
EOF
```

Включить и запустить системный демон gis:

```
systemctl enable gis --now
systemctl status gis
```

11.4. Установка и настройка модулей BackOffice

Распаковать дистрибутив:

```
tar xf /srv/distr/metasfera/backofficexd.tar.xz -C /opt
```

Создать файл настроек */opt/backofficexd/app_settings.json*:

```
cat <<EOF >/opt/backofficexd/app_settings.json
{
    "db_settings_model_3d": {
        "host": "127.0.0.1",
        "port": 5432,
        "user": "osm",
        "pwd": "<pg_pass>",
        "db_name": "model_3d"
    },

```

```
"db_settings_bim": {
  "host": "127.0.0.1",
  "port": 5432,
  "user": "osm",
  "pwd": "<pg_pass>",
  "db_name": "bim"
},
"db_settings_bim_models": {
  "host": "127.0.0.1",
  "port": 5432,
  "user": "osm",
  "pwd": "<pg_pass>",
  "db_name": "bim_models"
},
"db_settings_model_processing": {
  "host": "127.0.0.1",
  "port": 5432,
  "user": "osm",
  "pwd": "<pg_pass>",
  "db_name": "model_processing"
},
"auth_settings": {
  "host": "",
  "port": 0,
  "pmc_auth": true,
  "pmc_validate": true,
  "validate_token": false
},
"pmc_settings": [
  {
    "host": "<PMC_HOST>",
    "port": <PMC_PORT>,
    "use_project_tree": false
  }
]
```

```
    }
  ],
  "log_settings": {
    "exception_in_resp": true,
    "use_stdout": true,
    "level": "info",
    "product": "backofficexd"
  },
  "uvicorn_settings": {
    "host": "0.0.0.0",
    "port": 4508
  },
  "redis_settings": {
    "host": "127.0.0.1",
    "port": 6379,
    "db_index": 0,
    "cache_prefix": "fastapi_cache",
    "cache_namespace": "bim",
    "ttl": 86400,
    "enabled": true,
    "update_time": 14400,
    "resolve_host": "127.0.0.1"
  },
  "rabbitmq_settings": {
    "host": "127.0.0.1",
    "port": 5672,
    "user": "admin",
    "pwd": "<rmq_pass>",
    "callback_queue": "output_data",
    "publish_queue": "input_data",
    "blocked_connection_timeout": 65535,
    "heartbeat": 65535,
    "virtual_host": "/"
  }
}
```

```
    },
    "nestjs_settings": {
      "host": "127.0.0.1",
      "port": 5002
    },
    "bim_models_db_api_settings": {
      "host": "127.0.0.1",
      "port": 5002,
      "version": "v1"
    },
    "processing_endpoint": {
      "host": "127.0.0.1",
      "port": 4508,
      "user": "<PMC_USER>",
      "pwd": "<PMC_PASS>"
    },
    "minio_server_settings": {
      "host": "127.0.0.1",
      "port": 9000,
      "access_key": "<miniouser>",
      "secret_key": "<miniopass>",
      "secure": false
    },
    "web_service_model_processing_settings": {
      "host": "0.0.0.0",
      "port": 4509
    },
    "encrypted_config": false
  }
}
EOF
```

Создать сервис системного демона backofficexd:

```
cat <<EOF >/etc/systemd/system/backofficexd.service
[Unit]
```

```
Description=Backoffice service
```

```
After=network.target
```

```
[Service]
```

```
User=root
```

```
WorkingDirectory=/opt/backofficexd
```

```
LimitNOFILE=4096
```

```
ExecStart=/opt/backofficexd/backofficexd
```

```
Restart=on-failure
```

```
RestartSec=5s
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
EOF
```

Включить и запустить системный демон backofficexd:

```
systemctl enable backofficexd --now
```

```
systemctl status backofficexd
```

Создать сервис системного демона backofficexd_master_process_consumers:

```
cat
```

```
<<EOF >/etc/systemd/system/backofficexd_master_process_consumers.service
```

```
[Unit]
```

```
Description=backofficexd_master_processing_consumers
```

```
After=network.target
```

```
[Service]
```

```
User=root
```

```
WorkingDirectory=/opt/backofficexd
```

```
LimitNOFILE=4096
```

```
ExecStart=/opt/backofficexd/backofficexd_master_process_consumers
```

```
Restart=on-failure
```

```
RestartSec=5s
```

```
[Install]
WantedBy=multi-user.target
EOF
```

Включить и запустить системный демон

backofficexd_master_process_consumers:

```
systemctl enable
backofficexd_master_process_consumers.service --now
systemctl status
backofficexd_master_process_consumers.service
```

Создать сервис системного демона

backofficexd_model_processing_consumers:

```
cat
<<EOF >/etc/systemd/system/backofficexd_model_processing_co
nsumers.service
[Unit]
Description=backofficexd_model_processing_consumers
After=network.target

[Service]
User=root
WorkingDirectory=/opt/backofficexd
LimitNOFILE=4096
ExecStart=/opt/backofficexd/backofficexd_model_processing_c
onsumers
Restart=on-failure
RestartSec=5s

[Install]
WantedBy=multi-user.target
EOF
```

Включить и запустить системный демон

backofficexd_model_processing_consumers:

```
systemctl enable
backofficexd_model_processing_consumers.service --now
systemctl status
backofficexd_model_processing_consumers.service
```

11.5. Установка и настройка модуля Etl

Распаковать дистрибутив:

```
tar xf /srv/distr/metasfera/etl.tar.xz -C /opt
```

Создать файл настроек */opt/etl/app_settings.json*:

```
cat <<EOF >/opt/etl/app_settings.json
{
    "db_settings": {
        "host": "127.0.0.1",
        "port": 5432,
        "user": "osm",
        "pwd": "<pg_pass>",
        "db_name": "etl"
    },
    "auth_settings": {
        "host": "",
        "port": 0,
        "pmc_auth": true,
        "pmc_validate": true,
        "validate_token": false
    },
    "pmc_settings": [
        {
            "host": "<PMC_HOST>",
            "port": <PMC_PORT>,
            "use_project_tree": false
        }
    ]
}
```

```
],
  "log_settings": {
    "exception_in_resp": true,
    "use_stdout": true,
    "level": "info",
    "product": "etl"
  },
  "uvicorn_settings": {
    "host": "0.0.0.0",
    "port": 4506
  },
  "bim_settings": {
    "host": "127.0.0.1",
    "port": 4508
  },
  "rabbitmq_settings": {
    "host": "127.0.0.1",
    "port": 5672,
    "user": "admin",
    "pwd": "<rmq_pass>",
    "virtual_host": "/"
  },
  "encrypted_config": false
}
```

EOF

Создать сервис системного демона etl:

```
cat <<EOF >/etc/systemd/system/etl.service
[Unit]
Description=Etl service
After=network.target

[Service]
User=root
```

```
WorkingDirectory=/opt/etl
LimitNOFILE=4096
ExecStart=/opt/etl/etl
```

```
Restart=on-failure
RestartSec=5s
```

```
[Install]
WantedBy=multi-user.target
EOF
```

Включить и запустить системный демон etl:

```
systemctl enable etl --now
systemctl status etl
```

Создать сервис системного демона etl-ksg-consumer:

```
cat <<EOF >/etc/systemd/system/etl-ksg-consumer.service
```

```
[Unit]
Description=Analytics KSG service
After=network.target
```

```
[Service]
User=root
WorkingDirectory=/opt/etl
LimitNOFILE=4096
ExecStart=/opt/etl/etl_ksg_consumer
Restart=on-failure
RestartSec=5s
```

```
[Install]
WantedBy=multi-user.target
EOF
```

Включить и запустить системный демон etl-ksg-consumer:

```
systemctl enable etl-ksg-consumer.service --now
```

```
systemctl status etl-ksg-consumer.service
```

11.6. Установка и настройка микросервисов

Распаковать дистрибутив:

```
tar xf /srv/distr/metasfera/microservices.tar.xz -C /opt
```

Создать файл */opt/microservices/ecosystem.config.js*:

```
cat <<EOF >/opt/microservices/ecosystem.config.js
module.exports = {
  apps : [
    {
      name      : 'micro-api-swagger',
      cwd       : 'api-swagger',
      script    : 'apps/api-swagger/main.js',
      max_memory_restart: '3G',
      max_old_space_size: '2192',
      env: {
        BACKEND_BIM_API_URL: 'http://127.0.0.1:4508',
        BACKEND_BIM_API_CREDENTIALS:
'110:<PMC_USER>:<PMC_PASS>',
        POSTGRES_URI:
'postgres://osm:<pg_pass>@localhost:5432/bim_models',
        S3_ACCESS_KEY: '<miniouser>',
        S3_SECRET_KEY: '<miniopass>',
        S3_URI: 'http://127.0.0.1:9000'
      },
    },
  ],
  {
    name      : 'micro-api-websocket-gateway',
    cwd       : 'api-websocket-gateway',
    script    : 'apps/api-websocket-gateway/main.js',
    max_memory_restart: '9G',
    max_old_space_size: '8192',
    env: {
```

```
        POSTGRES_URI:
'postgres://osm:<pg_pass>@localhost:5432/bim_models',
        REDIS_URI: 'redis://127.0.0.1:6379/0'
    },
},
{
    name      : 'micro-compare-collision',
    cwd       : 'compare-collision',
    script    : 'apps/compare-collision/main.js',
    max_memory_restart: '9G',
    max_old_space_size: '8192',
    env: {
        AMQP_URI:
'amqp://admin:<rmq_pass>@127.0.0.1:5672/',
        POSTGRES_URI:
'postgres://osm:<pg_pass>@localhost:5432/bim_models',
        S3_ACCESS_KEY: '<miniouser>',
        S3_SECRET_KEY: '<miniopass>',
        S3_URI: 'http://127.0.0.1:9000'
    },
},
{
    name      : 'micro-glb-construction-stage-generator',
    cwd       : 'glb-construction-stage-generator',
    script    : 'apps/glb-construction-stage-generator/main.js',
    max_memory_restart: '16G',
    max_old_space_size: '8192',
    env: {
        AMQP_URI:
'amqp://admin:<rmq_pass>@127.0.0.1:5672/',
        POSTGRES_URI:
'postgres://osm:<pg_pass>@localhost:5432/bim_models',
```

```
        S3_ACCESS_KEY: '<miniouser>',
        S3_SECRET_KEY: '<miniopass>',
        S3_URI: 'http://127.0.0.1:9000'
    },
},
{
    name      : 'micro-ifc-nwd-converter',
    cwd       : 'ifc-nwd-converter',
    script    : 'apps/ifc-nwd-converter/main.js',
    max_memory_restart: '9G',
    max_old_space_size: '8192',
    env: {
        AMQP_URI:
'amqp://admin:<rmq_pass>@127.0.0.1:5672/',
        POSTGRES_URI:
'postgres://osm:<pg_pass>@localhost:5432/bim_models',
        S3_ACCESS_KEY: '<miniouser>',
        S3_SECRET_KEY: '<miniopass>',
        S3_URI: 'http://127.0.0.1:9000'
    },
},
{
    name      : 'micro-mlt-cde-nwd-tridb-importer',
    cwd       : 'mlt-cde-nwd-tridb-importer',
    script    : 'apps/mlt-cde-nwd-tridb-importer/main.js',
    max_memory_restart: '9G',
    max_old_space_size: '8192',
    env: {
        AMQP_URI:
'amqp://admin:<rmq_pass>@127.0.0.1:5672/',
        POSTGRES_URI:
'postgres://osm:<pg_pass>@localhost:5432/bim_models',
        S3_ACCESS_KEY: '<miniouser>',
```

```
        S3_SECRET_KEY: '<miniopass>',
        S3_URI: 'http://127.0.0.1:9000'
    },
},
{
    name      : 'micro-xls-defects-importer',
    cwd       : 'xls-defects-importer',
    script    : 'apps/xls-defects-importer/main.js',
    max_memory_restart: '9G',
    max_old_space_size: '8192',
    env: {
        AMQP_URI:
'amqp://admin:<rmq_pass>@127.0.0.1:5672/',
        POSTGRES_URI:
'postgres://osm:<pg_pass>@localhost:5432/bim_models',
        S3_ACCESS_KEY: '<miniouser>',
        S3_SECRET_KEY: '<miniopass>',
        S3_URI: 'http://127.0.0.1:9000'
    },
},
]
}
EOF
```

Запустить микросервисы:

```
cd /opt/microservices
pm2 start
pm2 list
pm2 save
```

11.7. Конфигурация веб-сервера

Создать директорию для размещения web:

```
mkdir /var/www
```

В файл `/etc/nginx/nginx.conf` добавить настройку обработки файлов типа `mjs`:

```
types {
    application/javascript mjs;
}
```

Пример настройки:

```
types_hash_max_size 4096;

include /etc/nginx/mime.types;
types {
    application/javascript mjs;
}
default_type application/octet-stream;
```

11.7.1. Установка и настройка модуля Metasfera xD.GIS

Распаковать файлы модуля Metasfera xD.GIS:

```
tar xf /srv/distr/metasfera/gis-web.tar.xz -C /var/www
```

Создать файл `/etc/nginx/conf.d/gis-web.conf`:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    server_name _;
    root /var/www/gis-web;

    client_max_body_size 0;
    underscores_in_headers on;

    proxy_read_timeout 180;
    proxy_http_version 1.1;

    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For
    $proxy_add_x_forwarded_for;
```

```
proxy_set_header X-Forwarded-Proto $scheme;

location = /assets/module-federation.manifest.json {
    add_header 'Content-Type' 'application/json';
    return 200 '{
"gis_2_0": "http://<server_ip>:81",
"xd_2_0": "http://<server_ip>:82",
"gis": "http://<server_ip>:82"
}';
    expires -1;
}

location / {
    try_files $uri $uri/ =404;
    add_header 'Access-Control-Allow-Origin' '*';
}

location = /index.html {
    add_header 'Access-Control-Allow-Origin' '*';
expires -1;
}

location = /remoteEntry.mjs {
    add_header 'Access-Control-Allow-Origin' '*';
expires -1;
}

location /api/ {
    proxy_pass http://<PMC_HOST>:<PMC_PORT>;
}

location ~ ^/(3d-assets|storage)/ {
    proxy_pass http://127.0.0.1:9000;
}

location /assets/models_3d/ {
```

```
        proxy_pass http://127.0.0.1:9000/3d-
assets/models_3d/;
    }

    location /geoserver/ {
        proxy_pass http://127.0.0.1:8080;
    }
    location /metasfera/ {
        proxy_pass http://127.0.0.1:81;
    }

    location /gis/ {
        proxy_pass http://127.0.0.1:4500;
    }
    location ~ ^/(media|static_sa)/ {
        proxy_pass http://127.0.0.1:4502;
    }
    location /etl/ {
        proxy_pass http://127.0.0.1:4506;
    }
    location /dp/ {
        proxy_pass http://127.0.0.1:4508;
    }
    location /3d-geometry-backend.socket.io/ {
        proxy_pass http://127.0.0.1:5000;
        proxy_set_header Upgrade      $http_upgrade;
        proxy_set_header Connection 'upgrade';
    }
    location /joints/ {
        proxy_pass http://127.0.0.1:5002/;
    }
}
```

11.7.2. Установка и настройка Shell

Распаковать файлы Shell:

```
tar xf /srv/distr/metasfera/shell.tar.xz -C /var/www
```

Создать файл */etc/nginx/conf.d/shell.conf*:

```
server {
    listen      81 default_server;
    listen      [::]:81 default_server;
    server_name _;
    root        /var/www/shell;

    client_max_body_size 0;
    underscores_in_headers on;

    proxy_read_timeout 180;
    proxy_http_version 1.1;

    proxy_set_header Host                $http_host;
    proxy_set_header X-Real-IP           $remote_addr;
    proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    location = /metasfera/assets/module-
federation.manifest.json {
        add_header 'Content-Type' 'application/json';
        return 200 '{
"legacy": "http://<server_ip>:80",
"gis": "http://<server_ip>:82"
}';
        expires -1;
    }
}
```

```
location / {
    try_files $uri $uri/ =404;
    add_header 'Access-Control-Allow-Origin' '*';
}
location = /index.html {
    add_header 'Access-Control-Allow-Origin' '*';
    expires -1;
}
location = /remoteEntry.mjs {
    add_header 'Access-Control-Allow-Origin' '*';
    expires -1;
}

location /assets/gis/ {
    proxy_pass http://127.0.0.1:80;
}
location /metasfera/geoserver/ {
    proxy_pass http://127.0.0.1:8080/geoserver/;
}

location /metasfera/gis/ {
    proxy_pass http://127.0.0.1:4500/gis/;
}
location /metasfera/media/ {
    proxy_pass http://127.0.0.1:4502/media/;
}
location /metasfera/static_sa/ {
    proxy_pass http://127.0.0.1:4502/static_sa/;
}
}
```

11.7.3. Установка и настройка модуля Metasfera xD: Metasfera Dynamics

Распаковать файлы модуля:

```
tar xf /srv/distr/metasfera/xd.tar.xz -C /var/www
```

Создать файл */etc/nginx/conf.d/xd.conf*:

```
server {
    listen      82 default_server;
    listen      [::]:82 default_server;
    server_name _;
    root        /var/www/xd;

    client_max_body_size 0;
    underscores_in_headers on;

    proxy_read_timeout 180;
    proxy_http_version 1.1;

    proxy_set_header Host                $http_host;
    proxy_set_header X-Real-IP           $remote_addr;
    proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    location / {
        try_files $uri $uri/ =404;
        add_header 'Access-Control-Allow-Origin' '*';
    }
    location = /index.html {
        add_header 'Access-Control-Allow-Origin' '*';
    }
    expires -1;
}
location = /remoteEntry.mjs {
```

```
        add_header 'Access-Control-Allow-Origin' '*';
    expires -1;
    }
}
```

11.7.4. Выполнение перезапуска Nginx

Сделать перезапуск nginx, чтобы перечитались конфигурационные файлы:

```
systemctl restart nginx
```

12. Запуск браузера

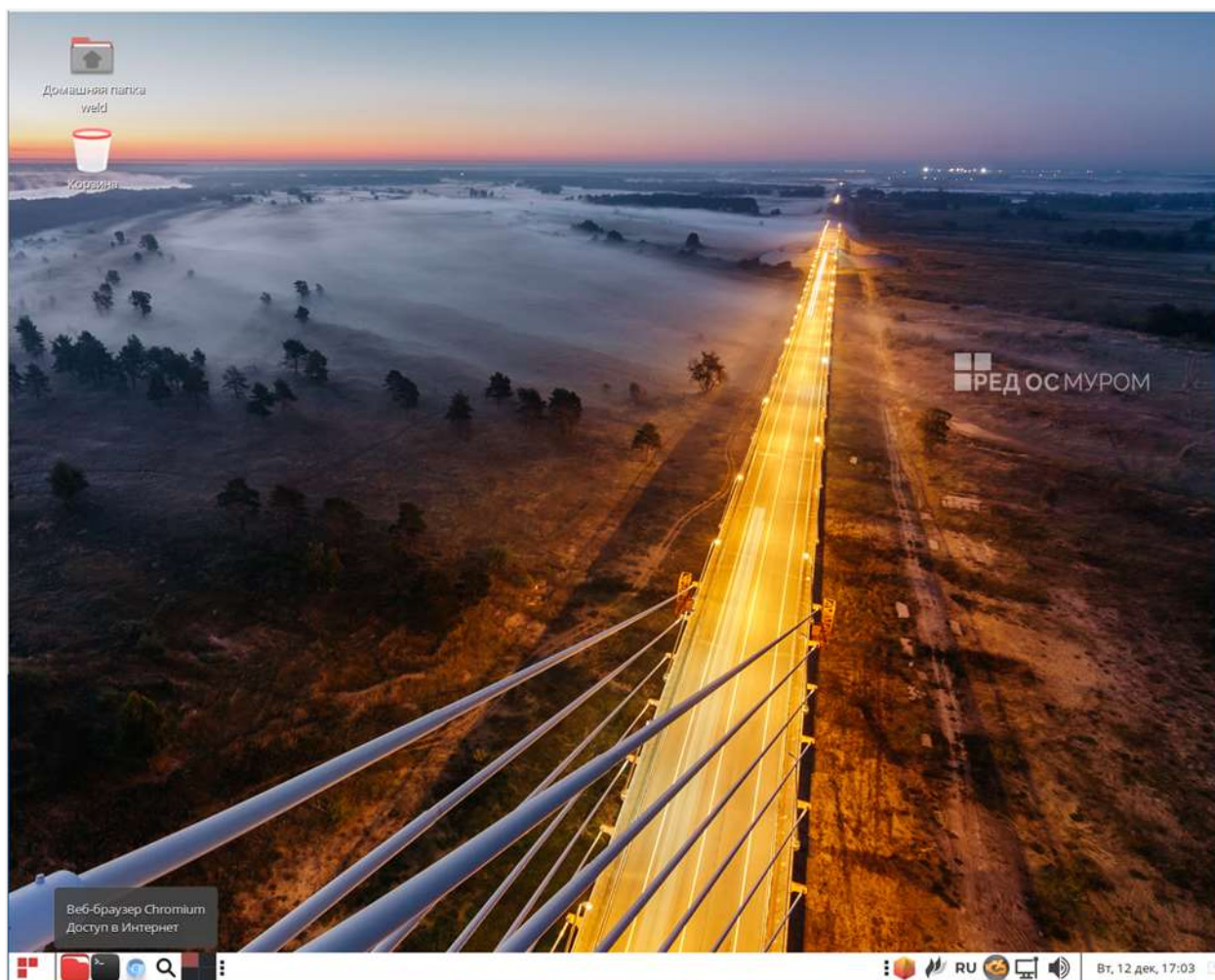


Рисунок 1. Рабочий стол

Необходимо нажать на иконку браузера (в данной операционной системе установлен по умолчанию «Chromium»), после чего браузер будет запущен (Рисунок 1).

В адресной строке необходимо ввести адрес `http://<server_ip>/`

После чего откроется страница логина. На странице логина необходимо ввести логин и пароль от системы (Рисунок 2).

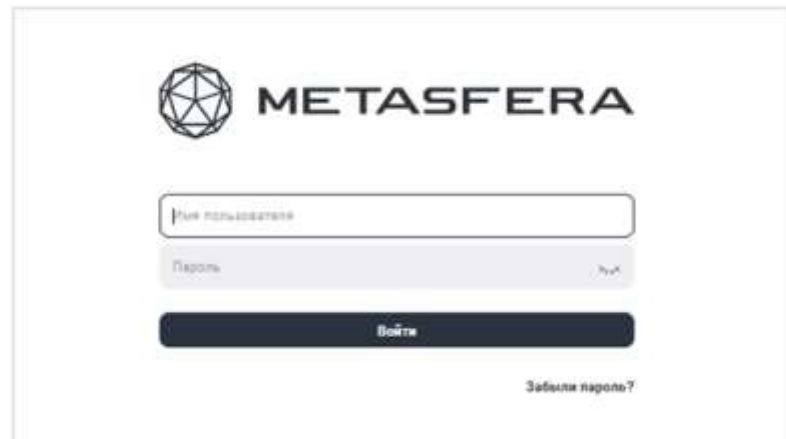


Рисунок 2. Страница входа