



ООО «Медицинские системы визуализации»  
197110, Санкт-Петербург, Левашовский пр.,  
д.12, лит. А, пом. 29Н, ком. 7

+7 812 334 49 39  
[www.mvsystem.ru](http://www.mvsystem.ru)  
[mail@mvsystem.ru](mailto:mail@mvsystem.ru)

## Контрольно-информационная панель управления Lyra

Инструкция по установке

© 2023 ООО "Медицинские системы визуализации"



## **Настройка CentOS 7**

1.1 Настройка ОС

1.2 Настройка локальной сети

1.3 Настройка времени, установка часовых поясов (Chrony client)

1.4 Подключение репозитория MVS

1.5 Настройка SSH на сервере

## **2 Настройка Ubuntu**

2.1 Настройка BIOS

2.2 Настройка ОС

2.3 Настройка локальной сети

2.6 Установка nginx Ubuntu

2.7 Установка postgresql Ubuntu

## **3. Сервер КИПУ Lyra**

3.1 Подготовка

3.2 Установка ПО

3.2.1 Добавление репозитория

3.3 Настройка Control API

4.1 Подготовка

4.2 Разметка дисков

4.4 Настройка звука ALSAMIXER

## **5. Главный монитор управления (GMU)**

5.1 Общая конфигурация

5.2 - Настройка приложений (секция apps)

5.4 - Обзорный монитор (VIEW)

## **6. Инженерная Система**

6.1 Установка сервера

6.2 Установка сервера на Ubuntu

6.3 - Установка клиента

## **7. Коридорный монитор (Intercom)**

## **Введение**

Настоящий документ является руководством по установке программного обеспечения, входящего в состав программно-аппаратного комплекса Контрольно-информационная панель MVS Lyra (КИПУ Lyra) (далее по тексту - ПО).

Руководство по установке ПО содержит перечень требований и файлов, необходимых для установки и работы ПО и определяет порядок действий персонала, необходимые для установки ПО.

ПО рассчитано на круглосуточную работу в течение всего года.

## **Квалификация персонала**

Лица, допущенные к работе с ПО (медицинские работники), должны быть ознакомлены с правилами техники безопасности, правилами эксплуатации технических средств, основного и вспомогательного оборудования. Специфических технических знаний не требуется.

## **Компоненты**

ПО КИПУ Lyra состоит из 2 компонентов:

Web-приложение, обеспечивающее работу пользователя с функциями операционного терминала;

Сервер, обеспечивающий работу приложения.

**Обратите внимание! Программное обеспечение не работает без аппаратной части!**

## **Перечень требований, необходимых для установки и работы ПО**

Характеристики информационной панели для КИПУ Lyra:

№	Характеристика	Значение
1	Корпус	UCFF (Ultra Compact Form Factor)

2	Процессор	Intel Core 7-го поколения и выше или с аналогичными характеристиками
3	Оперативная память	Объём не менее 8 Гбайт
4	Долговременная память	Объём не менее 250 Гбайт
5	Разъём USB	Не менее одного
6	Сетевой разъём	RJ45  Скорость передачи данных 1 Гбит/с
7	Видеовыход	HDMI
8	Аудиовыход	Наличие
9	Блок питания	19 В
10	Монитор	Диагональ не менее 20 дюймов  Разрешение не менее 1920x1280
11	Динамик	Наличие

Характеристики сервера для КИПУ Lyra:

№	Характеристика	Значение
1	Корпус	Для монтирования в серверную стойку (rackmount)

2	Процессор	Intel Xeon E3-1220 v6 LGA 1151 8Mb 3.0Ghz или с аналогичными характеристиками
3	Оперативная память	Объём не менее 8 Гбайт
4	Долговременная память для программы	SSD NVMe Объём не менее 250 Гбайт
5	Долговременная память для данных	SSD Объём не менее 240 Гбайт
6	Сетевой разъём	RJ45  Скорость передачи данных не менее 1 Гбит/с  Не менее 2 шт.
7	Блок питания	400 Вт

## Подготовка к установке

### Настройка CentOS 7

#### 1.1 Настройка ОС

Обновить систему:

```
yum -y update && yum -y upgrade
```

Отключить SELinux:

```
setenforce 0
```

```
cat > /etc/selinux/config <<EOF  
SELINUX=permissive  
SELINUXTYPE=targeted  
EOF
```

Подключить общий репозиторий и поставить утилиты:

```
yum -y install epel-release && yum -y update  
  
yum -y install bash-completion htop mc net-tools screen bind-utils vim deltarpm nano  
haveged chrony iftop ethtool wget curl lm_sensors yum-utils  
  
systemctl enable --now haveged  
  
systemctl disable postfix  
  
systemctl enable --now chronyd
```

Настроить имя хоста:

```
hostnamectl set-hostname <pc_name>.or<ip_address>.mvs.lyra  
systemctl restart systemd-hostnamed.service
```

Цветовая схема в консоли:

```
echo "export PS1='\[\033[00;33m\]\u\[\033[00;32m\]@\H\[\033[00m\]\[\033[01;34m\] \w  
\$\[\033[00m\] "' >> /home/mvsadmin/.bash_profile
```

```
echo "export PS1='\[\033[00;31m\]\u\[\033[00;32m\]@\H\[\033[00m\]\[\033[01;34m\] \w\n#\[\033[00m\] "' >> /root/.bash_profile

source .bash_profile
```

## 1.2 Настройка локальной сети

```
echo "net.ipv4.ip_forward = 1" > /etc/sysctl.conf
echo "net.ipv6.conf.all.disable_ipv6=1" > /etc/sysctl.d/80-ipv6.conf
sysctl -p
```

через консоль:

```
nmcli device status
nmcli connection show
nmcli connection modify <connectionName>/<mask>
nmcli connection modify <connectionName> ipv4.addresses <staticIp>
nmcli connection modify <connectionName> ipv4.gateway <gatewayIp>
nmcli connection modify <connectionName> ipv4.dns <dnsIp>
nmcli connection modify <connectionName> ipv4.method manual
nmcli connection modify <connectionName> ipv6.method ignore

nmcli connection up <connectionName>

nmcli connection <connection_name> autoconnect yes
```

или графическую утилиту:

```
Nmtui
```

на OR - настройка на ip-адрес сегмента + на ip-адрес больничной сети.

### 1.3 Настройка времени, установка часовых поясов (Chrony client)

На серверах под управлением CentOS - в BIOS ставить время в формате UTC (без часовых поясов)

Настроить NTP-сервер.

```
yum install chrony
```

Принудительно заставить chrony использовать только ipv4:

```
cat > /etc/sysconfig/chronyd << EOF
OPTIONS="-4"
EOF
```

Править файл конфигурации:

```
# vim /etc/chrony.conf
```

Сервер-конфиг:

```
bindcmdaddress <main_ip_HP>

# Allow NTP client access from local network.
# mask 21 == 64.0-71.0
# mask 20 == 64.0-79.0
allow 192.168.66.0/24
```

Клиент-конфиг: комментировать основные сервера и добавить адрес HP:

```
# server 3.centos.pool.ntp.org iburst
server <ip_HP> iburst
```

Запустить сервис и настроить часовую зону:



```
systemctl enable chronyd && systemctl restart chronyd
```

```
chronyc sources
```

```
timedatectl
```

```
timedatectl list-timezones
```

```
timedatectl set-timezone <ZONE>
```

Открыть порт firewalld, если настроен как сервер:

```
sudo firewall-cmd --permanent --zone=public --add-service=ntp
```

#### 1.4 Подключение репозитория MVS

```
sudo cat << EOF > /etc/yum.repos.d/mvs-yukki1.0.repo
```

```
[yukki1.0]
```

```
baseurl = https://<login>:<password>@repo.mvslyra.com/repository/yukki/
```

```
gpgcheck = 0
```

```
name = MVS yukki1.0
```

```
enabled = 1
```

```
[lyra3.3rdparty]
```

```
baseurl = https://<login>:<password>@repo.mvslyra.com/repository/lyra3.3rdparty/
```

```
gpgcheck = 0
```

```
name = MVS 3rdparty
```

```
enabled = 1
```

```
EOF
```

Выполнить команду:

```
sudo yum update
```

### 1.5 Настройка SSH на сервере

Необходимо сгенерировать ключевую пару для подключения к серверу (на локальной машине) и добавить на удаленный сервер:

```
ssh-keygen -t ed25519 -C "mvsadmin key" -N "" -f serverHostName.ec.privateKey
```

```
ssh-copy-id -i ~/.ssh/serverHostName.ec.pub mvsadmin@remote_ip
```

На удаленном сервере изменить имя файла, права доступа:

```
mv /home/mvsadmin/.ssh/authorized_keys /home/mvsadmin/.ssh/authorizedKeys
```

```
chown mvsadmin:mvsadmin -R /home/mvsadmin/.ssh
```

```
chmod 0500 /home/mvsadmin/.ssh
```

```
chmod 0400 /home/mvsadmin/.ssh/authorizedKeys
```

Добавить новый порт для ssh в firewalld:

```
firewall-cmd --permanent --zone=public --add-port=41385/tcp  
firewall-cmd --reload
```

На сервере сделать бэкап /etc/ssh/sshd\_config:

```
cp /etc/ssh/sshd_config /etc/ssh/sshd_config.orig
```

**Обратите внимание**, что терминальную сессию при выполнении следующих команд не закрывать, пока не настроится ssh и не будет проверено подключение.

Остановить службу ssh:

```
systemctl stop sshd
```

Почистить файлы.

```
rm -fv /etc/ssh/ssh_host_*
```

Очистить файл конфигурации ssh:

```
:> /etc/ssh/sshd_config
```

Вставить новый конфиг:

```
vim /etc/ssh/sshd_config
```

**Здесь в конфигурации:** отключение доступа по паролю, разрешено подключение только для mvsadmin, новый порт, не использовать резолвинг dns.

Port 41385

AllowUsers mvsadmin

AllowGroups wheel

LogLevel INFO

#LogLevel DEBUG

SyslogFacility AUTH

AcceptEnv LC\_\* LANG

AddressFamily inet

ListenAddress 0.0.0.0

TCPKeepAlive yes

UseDNS no

Protocol 2

AllowAgentForwarding no

AllowTcpForwarding no

X11Forwarding no

PermitRootLogin no

PermitEmptyPasswords no

PermitTunnel no

PermitUserEnvironment no

UsePAM yes

AuthorizedKeysFile .ssh/authorizedKeys

Banner none

PrintLastLog yes

```
PrintMotd no

ClientAliveCountMax 4
ClientAliveInterval 10
Compression yes
LoginGraceTime 60
MaxAuthTries 3
MaxSessions 8
MaxStartups 4
StrictModes yes
RekeyLimit 512M 1h

ChallengeResponseAuthentication no
GSSAPIAuthentication no
HostbasedAuthentication no
KerberosAuthentication no
PasswordAuthentication no
PubkeyAuthentication yes

HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ed25519_key

Subsystem sftp /usr/libexec/openssh/sftp-server
```

Сохранить и запустить службу:

```
systemctl start sshd
```

Проверить вход по ключу из новой консоли, чтобы была возможность что-то исправить.

```
ssh mvsadmin@remote_ip -p port -i path_to_priveteKey
```

## 2 Настройка Ubuntu

### 2.1 Настройка BIOS

В некоторых случаях требовалось перейти на ядро 5.6. Для этого предварительно необходимо отключить Security Boot на целевой машине. На Intel NUC в биосе необходимо отключить Thunderbolt, Wi-Fi, Bluetooth, SATA, кард-ридер и прочие неиспользуемые компоненты. Выбрать в разделе Video видеовыход по умолчанию (HDMI), иначе при включении с неподключенным HDMI изображения не будет.

### 2.2 Настройка ОС

Настройка имени ПК

```
hostnamectl set-hostname pc_name.or1-2-3-4.mvs.lyra
```

Установить необходимый инструментарий для начала установки:

```
apt update && apt upgrade -y && apt -y install wget curl gnupg vim net-tools xterm software-properties-common ca-certificates
```

### 2.3 Настройка локальной сети

Настройка сети производится в соответствии с общей схемой назначения ip-адресов.

Сервер операционной изолирует все компоненты внутри операционной от внешней (больничной) локальной сети и является шлюзом.

Внутренний сегмент:

OR1 - 192.168.66.0/24

OR2 - 192.168.67.0/24

и далее.

При назначении ip-адресов см. таблицу типовых адресов.

Шлюз - всегда ip-сервер операционной.

## 2.4 Настройка времени, установка часовых поясов (Chrony)

выключаем стандартную службу времени:

```
systemctl stop systemd-timesyncd && systemctl disable systemd-timesyncd
```

ставим chrony:

```
# apt -y install chrony
```

```
# vim /etc/chrony/chrony.conf
```

```
# комментируем все сервера и добавляем
```

```
pool {{ HP_EXT }} iburst maxsources 1
```

Перезапустить службу:

```
systemctl enable chrony && systemctl restart chrony
```

```
chronyc sources
```

```
timedatectl
```

```
timedatectl list-timezones
```

```
timedatectl set-timezone <ZONE>
```

Отключаем IPv6 для chrony:

```
cat > /etc/sysconfig/chronyd << EOF
```

```
OPTIONS="-4"
```

```
EOF
```

## 2.5 Подключение репозитория MVS

```
sudo apt install curl vim sudo gnupg -y
```

Добавить параметры авторизации в репозитории:

```
cat > /etc/apt/auth.conf.d/mvs_lyra.conf <<EOF
machine repo.mvslyra.com
login <user>
password <password>
EOF
```

Добавить файл репозитория в менеджер пакетов:

```
echo "deb [arch=amd64] deb https://repo.mvslyra.com/repository/yukki-deb bionic main" >
/etc/apt/sources.list.d/mvs_lyra3.list

echo "deb [arch=amd64] deb https://repo.mvslyra.com/repository/common-deb bionic
main" > /etc/apt/sources.list.d/mvs_lyra3.list

echo "deb [arch=amd64] deb https://repo.mvslyra.com/repository/lyra3.3rdparty-deb bionic
main" >> /etc/apt/sources.list.d/mvs_lyra3.list
```

Скачать и установить ключ подписи репозитория:

```
curl -L https://repo.mvslyra.com/keys/mvs_key.gpg | sudo apt-key add -
sudo apt update
```

## 2.6 Установка nginx Ubuntu

Требуется для конфигурации в случае ручной установки:

- GERS;
- сервера-клиента инженерных систем на одном ПК.

Все действия под выполняются sudo.



Подключение репозитория для стабильной ветки:

```
echo "deb [arch=amd64] http://nginx.org/packages/ubuntu `lsb_release -cs` nginx" \  
  | sudo tee /etc/apt/sources.list.d/nginx.list  
  
wget -qO - https://nginx.org/keys/nginx_signing.key | sudo apt-key add -
```

Установка:

```
apt update  
apt install nginx -y  
  
systemctl start nginx
```

## 2.7 Установка postgresql Ubuntu

Настройка репозитория:

```
echo "deb http://apt.postgresql.org/pub/repos/apt bionic-pgdg main" \  
  | sudo tee /etc/apt/sources.list.d/pgdg.list  
  
wget -qO - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
```

Установка и старт:

```
sudo apt install postgresql-15  
  
sudo enable --now postgresql
```

Настройка параметров:

```
# cat /etc/postgresql/15/main/conf.d/example.conf  
  
max_connections = 1000 # (change requires restart)  
  
shared_buffers = 512MB # min 128kB  
log_timezone = 'Europe/Moscow'
```

```
timezone = 'Europe/Moscow'  
password_encryption = scram-sha-256 # md5 or scram-sha-256  
  
# systemctl restart postgresql
```

### 3. Сервер КИПУ Lyra

#### 3.1 Подготовка

Настройка Firewalld:

```
systemctl disable iptables  
systemctl stop iptables  
systemctl enable firewalld  
systemctl start firewalld
```

Маршрутизация трафика:

```
firewall-cmd --zone=external --change-interface=<eth_ext>  
  
firewall-cmd --zone=trusted --change-interface=<eth_int>
```

eth\_int - внутренний интерфейс

eth\_ext - внешний интерфейс

Подробнее про использование зон и обработку трафика по умолчанию:

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7/html/security\\_guide/sec-working\\_with\\_zones#doc-wrapper](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sec-working_with_zones#doc-wrapper)

```
firewall-cmd --reload
```

## 3.2 Установка ПО

### 3.2.1 Добавление репозитория

Выполнить пункт [1.5 - Подключение репозитория MVS](#)

Проверить пакеты в репозитории можно с помощью команды:

```
sudo yum search mvs
```

#### Автоматическая установка

```
yum install mvs-lyra3-or-server -y
```

Заполнить параметры файла конфигурации:

```
vi /etc/mvs.lyra3.or/options.yml
```

Запустить скрипт конфигурирования:

```
mvsllyra3orctl setup
```

Проверить статус сервисов:

```
systemctl status Lyra3.OR.*
```

#### Ручная установка

Установка и настройка Postgres:

```
yum -y install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

```
yum -y install postgresql13-server postgresql13-contrib
```

Инициализация БД в каталоге по умолчанию `/var/lib/pgsql/13/data/`:

```
/usr/pgsql-13/bin/postgresql-13-setup initdb
```

Сгенерировать сертификаты, выставить права и владельца на них:

```
#openssl req -new -x509 -days 17945 -nodes -out /var/lib/pgsql/13/data/server.crt -keyout  
/var/lib/pgsql/13/data/server.key -subj "/CN=<serverHostName>"  
#chmod 0400 /var/lib/pgsql/13/data/server.{crt,key}  
#chown postgres:postgres /var/lib/pgsql/13/data/server.{crt,key}
```

где `<serverHostName>` текущее имя сервера

Редактировать (включить ssl-шифрование) `/var/lib/pgsql/13/data/postgresql.conf`:

```
listen_addresses = '127.0.0.1'  
max_connections = 1000  
ssl = on  
shared_buffers = 512MB  
password_encryption = scram-sha-256
```

Отредактировать `/var/lib/pgsql/13/data/pg_hba.conf`:

```
# IPv4 local connections:
host    all             all             127.0.0.1/32    scram-sha-256
hostssl <DatabaseName> <DatabaseUser> 0.0.0.0/0       scram-sha-256

# IPv6 local connections:
#host   all             all             ::1/128         scram-sha-256
#host   replication    all             ::1/128         scram-sha-256
```

В автозагрузку и старт сервиса:

```
systemctl enable --now postgresql-13
```

Изменить пароль для пользователя postgres (<postgresPassword>):

`/var/lib/pgsql/13/data/`

```
sudo -u postgres psql -c "ALTER USER postgres WITH ENCRYPTED PASSWORD
'<postgresPassword>'"
```

### 3.3 Настройка Control API

Control.API - функционал управления камерами PTZ через Streamer.

```
yum install -y mvs-lyra3-or-control-api
```

Редактировать:

```
vim /etc/mvs.lyra3.or/Lyra3.OR.Control.API.yml
```

Пример файла конфигурации:

*/etc/mvs.lyra3.or/Lyra3.OR.Control.API.yml:*

```
LoggerOptions:
```

```
# Trace | Debug | Information | Warning | Error | Critical | None
```

```
LogLevel: Trace
```

```
# Verbose | Full | Short | Tiny | Systemd
```

```
LogFormat: Systemd
```

```
#UseColors: false
```

```
ListenOptions:
```

```
DmzEndpoints:
```

```
- 0.0.0.0:3336
```

```
PublicEndpoints:
```

```
- 0.0.0.0:3337
```

```
PublicKeyRepositoryOptions:
```

```
PublicKeyPem: |-
```

```
-----BEGIN RSA PUBLIC KEY-----
```

```
-----END RSA PUBLIC KEY-----
```

```
LyraORServerOptions:
```

```
# '<value, like as 'http://server.url:port/', must be specified>'
```

```
Uri: 'http://127.0.0.1:3319'
```

Запуск службы:

```
systemctl enable --now Lyra3.OR.Control.API.service
```

Настроить Firewall:

```
firewall-cmd --zone=external --add-port=3336/tcp --permanent
```

```
firewall-cmd --reload
```

## 4. GERS

### 4.1 Подготовка

Выполнить предварительную настройку GERS, используя раздел [Настройка Ubuntu](#) с обязательным выполнением пункта [1.5 - Подключение репозитория MVS](#)

Для GERS требуется обеспечить аппаратное ускорение (VAAPI).

В случае использования процессоров **Intel ниже 10** серии необходимо установить драйвер:

```
sudo apt install i965-va-driver -y
```

В случае использования процессоров **Intel 10 серии и выше** необходимо установить драйвер:

```
sudo apt install intel-media-va-driver -y
```

После установки драйвера необходимо проверить работоспособность VAAPI с помощью утилиты vainfo. Пример вывода в случае работоспособной конфигурации:

...

VAProfileH264ConstrainedBaseline: VAEntrypointVLD

VAProfileH264ConstrainedBaseline: VAEntrypointEncSlice

VAProfileH264ConstrainedBaseline: VAEntrypointEncSliceLP

VAProfileH264Main : VAEntrypointVLD

VAProfileH264Main : VAEntrypointEncSlice

VAProfileH264Main : VAEntrypointEncSliceLP

VAProfileH264High : VAEntrypointVLD

VAProfileH264High : VAEntrypointEncSlice

VAProfileH264High : VAEntrypointEncSliceLP

VAProfileH264MultiviewHigh : VAEntrypointVLD

VAProfileH264MultiviewHigh : VAEntrypointEncSlice

VAProfileH264StereoHigh : VAEntrypointVLD

VAProfileH264StereoHigh : VAEntrypointEncSlice



...

## 4.2 Разметка дисков

Разметка дисков (при условии, что два SSD накопителя необходимо собрать в массив):

```
mdadm --zero-superblock --force /dev/sd{a,b}

wipefs --all --force /dev/sd{a,b}

mdadm --create --verbose --force /dev/md0 -l 0 -n 2 /dev/sd{a,b}

pvcreate -y /dev/md0

vgcreate -y gersstorage /dev/md0

lvcreate -yl +100%FREE -n storage1 gersstorage

mkfs.ext4 /dev/gersstorage/storage1

uuid=`blkid | grep storage1 | grep -oE [-a-z0-9]{36}`

echo "UUID=$uuid /var/sdmedia ext4 defaults 0 0" >> /etc/fstab
```

Разметка дисков (при использовании SSD накопителя M.2):

```
lsblk #найти необходимый диск

pvcreate -y /dev/<disk>

vgcreate -y gersstorage /dev/<disk>

lvcreate -yl +100%FREE -n storage1 gersstorage
```

```
mkfs.ext4 /dev/gersstorage/storage1
```

```
echo "/dev/gersstorage/storage1 /var/lib/mvs.lyra3.or ext4 defaults 0 0" >> /etc/fstab
```

### 4.3 Установка GERS

#### Автоматическая установка

Выполнить команду:

```
sudo apt install mvs-lyra3-or-gers
```

Открыть файл `/etc/mvs.lyra3.or/gers-options.yml`:

```
sudo vim /etc/mvs.lyra3.or/gers-options.yml
```

Заполнить параметры файла конфигурации GERS, обязательно заменив параметры в полях со значением `<set_it_up>` на актуальные значения.:

Lyra3:

HP:

Auth:

Address: `<set_it_up>` # http[s]://`<Address>`[:Port]

Core:

Address: `<set_it_up>` # http[s]://`<Address>`[:Port]

Streamer:

Address: <set\_it\_up> # http[s]://<Address>[:Port]

Packhaus:

Address: <set\_it\_up> # http[s]://<Address>[:Port]

OR:

Locale: ru

Server:

Address: <set\_it\_up> ## ip address

Port: 3319

GERS:

Id: "<set\_it\_up>" # должен быть идентичен названию из словаря  
/etc/mvs.lyra3.or/gerses.yml

LogLevel: Information

LogFormat: Systemd

MediaPath: /var/lib/mvs.lyra3.or/gers # либо путь, по которому примонтирован  
медиа раздел, если точка монтирования отличается от /var/lib/mvs.lyra3.or

WebStaticPath: /usr/share/mvs.lyra3.or/Lyra3.OR.GERS.SPA

ApiHost: 127.0.0.1 ## для настройки nginx upstreams

ApiPort: 3320 ## для настройки nginx upstreams

Support4K: "Off" # или "On"

Codec: H264VAAP1

MjpegQuality: 75

Endpoints:

- 0.0.0.0:3320

- 0.0.0.0:3321

- 0.0.0.0:3322

- 0.0.0.0:3323

- 0.0.0.0:3324

- 0.0.0.0:3325

- 0.0.0.0:3326

- 0.0.0.0:3327

- 0.0.0.0:3328

- 0.0.0.0:3329

- 0.0.0.0:3330

- 0.0.0.0:3331

- 0.0.0.0:3332

- 0.0.0.0:3333

- 0.0.0.0:3334

- 0.0.0.0:3335

ChannelsNumber: 4 # должен соответствовать количеству каналов в словаре  
ChannelOptions (ниже)

ChannelOptions:

```
- Title: "Channel0" # Количество элементов в словаре
```

```
DeinterlaceOnOff: "Off" # по количеству каналов
```

```
...
```

```
GrabberType: Blackmagic ## Blackmagic or Magewell
```

```
GERS_API_HPKEY: "<set_it_up>" # должен быть предварительно создан в Master, раздел Equipments.
```

Для удобства заполнения конфигурации актуальные значения адресов серверов приведены в Master (разделе Components - далее необходимо нажать на интересующий компонент, требуемый адрес будет в поле «URL»).

Выполнить команду:

```
sudo mvslra3gersctl setup
```

После выполнения команды проверить статус работы сервисов, выполнив команду:

```
systemctl status Lyra3.OR.GERS.API
```

Выполнить перезагрузку системы:

```
reboot
```

После перезагрузки проверить работоспособность web-интерейса GERS, набрав в поле адреса браузера [http://<адрес\\_GERS>](http://<адрес_GERS>)

### Ручная установка

Установить ПО MVS:

```
apt install -y mvs-lyra3-or-gers-api mvs-lyra3-or-gers-spa
```

Проверить актуальность прошивки платы видеозахвата в случае использования платы Blackmagic:

```
/usr/lib/blackmagic/DesktopVideo/DesktopVideoUpdateTool -ua
```

Разделить порты на плате видеозахвата:

```
ssh -XYC mvsadmin@gers_ip
```

```
/usr/lib/blackmagic/DesktopVideo/BlackmagicDesktopVideoSetup
```

Отредактировать конфиг */etc/mvs.lyra3.or/Lyra3.OR.GERS.API.yml*:

```
LoggerOptions:
```

```
# Trace | Debug | Information | Warning | Error | Critical | None
```

```
LogLevel: Information
```

```
# Verbose | Full | Short | Tiny | Systemd
```

```
LogFormat: Verbose
```

```
UseColors: true
```

```
ListenOptions:
```

```
DmzEndpoints:
```

```
- 0.0.0.0:3320
```

```
- 0.0.0.0:3321
```

- 0.0.0.0:3322

- 0.0.0.0:3323

- 0.0.0.0:3324

- 0.0.0.0:3325

- 0.0.0.0:3326

- 0.0.0.0:3327

- 0.0.0.0:3328

- 0.0.0.0:3329

- 0.0.0.0:3330

- 0.0.0.0:3331

- 0.0.0.0:3332

- 0.0.0.0:3333

- 0.0.0.0:3334

- 0.0.0.0:3335

GrabberOptions:

MaxChannelsNumber: 4 ## Max channel to grab.

Substreams: ## Substreams config

- Name: "Original"

Width: 1920

Height: 1080

- Name: "High"

Width: 1280

Height: 720

- Name: "Med"

Width: 720

Height: 400

- Name: "Low"

Width: 352

Height: 192

Titles: # количество должно равняться MaxChannelsNumber (см выше)

- gers2\_Channel0

- gers2\_Channel1

- gers2\_Channel2

- gers2\_Channel3

## Support 4k for recording. Values: Off, On, Quadro. Default value: Off

Support4K: "Off"

## Deinterlace support per channel On, Off - default On



Deinterlace:

- "Off"

- "Off"

- "Off"

- "Off"

##Use fake grabber and software codecs

UseFake: false

GrabberType: Blackmagic

MjpegStreamerOptions:

## Quality of stream (range: 1-100)

Quality: 75

## DeviceName - name of device with VA-API capability

# DeviceName: "/dev/dri/renderD128"

RecorderOptions:

## Codec to use in encoding (available now H264CUDA, H264VA-API - enum type)

Codec: H264VA-API

```
## Bitrate to use video codec in kbit per second
```

```
Bitrate: 10000
```

```
## Path to root folder to record media
```

```
MediaPath: /var/lib/mvs.lyra3.org/gers
```

```
## Video fragment duration in seconds (use -1 to infinite duration, e.g. 1 segment per channel), 0 is not allowed and cause exception
```

```
FragmentDuration: 6
```

```
## DeviceName - name of device with hardware capability
```

```
# DeviceName: "/dev/dri/renderD128"
```

```
## AudioDeviceName - name of input audio device
```

```
AudioDeviceName: "hw:0,0"
```

```
## Disable fps restrictions
```

```
# DisableFpsRestrictions: false
```

```
## Ignore audio device if fail default: false
```

```
IgnoreAudioFail: false
```

```
ORServerOptions:
```

```
## Address and Port is used by default.
```

```
Address: <адрес оп сервера>
```

```
Port: 3319
```

```
## GERS id for interact with OR.Server.API. Allowed symbols are [a-zA-Z0-9]
```

```
GersId: gers2
```

```
HrPlatformOptions:
```

```
HrKey: <Берется из раздела Equipments в мастере>
```

```
AuthApiUri: <http://адрес hr:3304>
```

```
CoreApiUri: <http://адрес Lyra3.HP.Core.API:3302>
```

```
StreamerApiUri: <http://адрес Lyra3.HP.Streamer.API:3308>
```

```
PackhausApiUri: <http://адрес Lyra3.HP.Packhaus.API:3316>
```

```
SurgeryUploadServiceOptions:
```

```
MaxParallelUploads: 8 ## Количество одновременных загрузок в пакгауз от 1 до 8
```

**Внимание!** Для оптимизации занимаемого видеопотоком места на GERS, а в последствии и на дисках Packhaus необходимо раскомментировать поле конфигурации Bitrate: и указать корректное значение битрейта видеопотока.

Рекомендуемые значения для поля Bitrate при разрешении видеопотока 1080 – 10000 кбит/с.

При разрешении видеопотока 4k – 17000 кбит/с.

Отредактировать конфигурацию веб сервера */etc/nginx/nginx.conf*:

```
user www-data;  
error_log /var/log/nginx/error.log warn;
```

```
pid /run/nginx.pid;
worker_processes "auto";
events {
worker_connections 1000;
multi_accept off;
}
http {
include /etc/nginx/mime.types;
default_type application/octet-stream;
server_names_hash_bucket_size 64;
client_max_body_size 6400m;
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 buffer=16k flush=2m;
sendfile on;
tcp_nopush on;
tcp_nodelay on;
keepalive_timeout 65;
keepalive_requests 100;
server_tokens on;
gzip on;
include /etc/nginx/conf.d/*.conf;
}
```

Отредактировать конфигурацию веб-сервера */etc/nginx/conf.d/Lyra3.OR.GERS.SPA.conf*:

```
# cat /etc/nginx/conf.d/Lyra3.OR.GERS.SPA.conf

server {
```

```
listen 80 default;

location / {

    root /usr/share/mvs.lyra3.or/Lyra3.OR.GERS.SPA;

    index index.html;

    try_files $uri /index.html;

}

location ~ ^/gers/(.*)$ {

    set $GERS_API_PATH http://127.0.0.1:3320/$1$is_args$args;

    proxy_pass $GERS_API_PATH;

}

location /media {

    alias /var/lib/mvs.lyra3.or/gers;

    autoindex on;

    autoindex_exact_size on;

    autoindex_format html;

}
```

```
error_page 500 502 503 504 /50x.html;

location = /50x.html {

    root /usr/share/nginx/html;

}

}
```

Отредактировать конфиг # cat /usr/share/mvs.lyra3.or/Lyra3.OR.GERS.SPA/config.json:

```
{

"publicDir": "",

"apiUrl": "gers"

}:

}
```

Разделяем каналы.

Заходим на GERS с ключами “-XYC”: ssh -XYC mvsadmin@192.168.66.XX

Вызываем десктопную утилиту (**только не от рута!**) BlackMagic и выбираем одиночный порт на каждом канале:

```
/usr/lib/blackmagic/DesktopVideo/./BlackmagicDesktopVideoSetup
```

#### 4.4 Настройка звука ALSAMIXER

```
alsamixer
```

```
reboot
```

#### 5. Главный монитор управления (GMU)

Обновить кэш репозитория и установить приложение для настройки звука:

```
apt update && apt upgrade -y && apt install -y pavucontrol-qt
```

Установить ПО MVS для терминала ГМУ:

```
apt install -y mvs-lyra3-or-terminalapp
```

Отредактировать конфиг `/etc/mvs.lyra3.or/Lyra3.OR.TerminalApp.yml`:

##### 5.1 Общая конфигурация

- Секция `httpConnection`:

```
Lyra3.OR.Server URL – URL в виде http://адрес:порт, порт dmz-эндпоинта Lyra3.OR.Server
```

- Секция `snapshotsLocation` содержит единственный параметр, URL http сервера nginx.

- В секции `hpConnection` необходимо указать URLы nginx в следующем виде:

```
authApiUrl: "<Lyra3.HP nginx URL>/rest/auth-api"
```

```
coreApiUrl: "<Lyra3.HP nginx URL>/rest/core-api"
```

```
storageUrl: "<Lyra3.HP nginx URL>/storage"
```

```
jumboApiUrl: "<Lyra3.HP nginx URL>/rest/jumbo-api"
```

Подробнее о настройке nginx в разделе “Настройка госпитального nginx”

- В секции *authorization.hpKey* – ключ для оборудования, получаемый в Мастере.

- Секция *rabbitConnection.url*: url подключения к rabbitmq, обслуживающего очереди сервера операционной в виде:

```
amqp://<rabbitmq login>:<rabbitmq password>@<rabbitmq Address>:5672
```

- Секция *monitorId* должна содержать тот же идентификатор, что указан для соответствующего монитора в файле *monitors.yml* сервера операционной.

## 5.2 - Настройка приложений (секция apps)

Секция *apps* представлена в виде словаря:

```
apps:
```

```
- enabled: true
```

```
  name: sip
```

```
  ...
```

```
- enabled: true
```

```
  name: ovConference
```

```
  ...
```

```
- enabled: false
```

```
  name: esc1
```



```
...
```

Включить/выключить приложения в секции `apps` можно установив параметр `enabled`: в `true/false` соответственно.

### 5.3 Автозапуск графического сеанса

Настройка автозапуска для пользователя `mvs` выполняется в пакете `mvs-autologin`, который устанавливается как зависимость пакета `mvs-lyra3-or-terminalapp`.

В случае необходимости ручной настройки автозапуска предпринимаются следующие действия:

```
sudo su - mvs
cd ~
```

`vim .xsession`:

```
exec openbox-session
```

`vim .bash_profile`:

```
[[ -z $DISPLAY && $XDG_VTNR -eq 1 ]] && exec startx -- -nocursor >/dev/null 2>&1
```

Дополнить `/etc/xdg/openbox/rc.xml`

```
<keybind key="C-A-t">
  <action name="Execute">
    <startupnotify>
      <enabled>true</enabled><name>Starting xterm</name>
    </startupnotify>
    <command>xterm</command>
  </action>
```

```
</keybind>

<keybind key="C-A-BackSpace">
  <action name="Execute">
    <command>pkill X</command>
  </action>
</keybind>
```

Дополнить `/etc/xdg/openbox/autostart`:

```
xset s off
xset s noblank
xset -dpms

amixer sset Master unmute
amixer sset Headphone unmute
amixer sset Front unmute

amixer set Master 90%
amixer set Headphone 90%
amixer set Front 90%
```

Перезагрузить службу:

```
systemctl restart Lyra3.OR.TerminalApp.service
```

```
reboot
```

#### 5.4 - Обзорный монитор (VIEW)

Настройка обзорного монитора выполняется аналогично настройке ГМУ.

В настройках *monitors.yml* сервера операционной выставляем параметр *isTouch: false*

## 6. Инженерная Система

### 6.1 Установка сервера

Установка ПО на платформу сервера Операционной производится после настройки пакета для КИПУ Lyra.

```
yum -y install mvs-esc1-server-service-setup
```

В логе установки найти пароль и сохранить его для доступа в админ через браузер.

Изменить имя домена в nginx. Конфиг файл для nginx:

```
vim /etc/nginx/conf.d/ESC1.Server.Web.conf
```

```
server{  
  
...  
  
listen 443 ssl default_server;  
    # EDIT server_name  
    server_name <ИМЯ_ДОМЕНА>;  
  
    root /srv/MVS.ESC1.Server.Web;  
  
...  
}
```

Перезагрузить nginx:

```
nginx -s reload
```

Прописать сертификаты, аналогично пункту настройка госпитального nginx.

В файле `/etc/nginx/conf.d/mvs.default_http_with_redirect.conf` поменять порт на 8081 и перезагрузить nginx.

Проверить состояние службы:

```
systemctl status ESC1.Server.API.service
```

Чтобы двери имели функцию блокировки:

Редактируем `/etc/mvs.esc1/plc.config.default.yml`

В ветке `Doors` выставить `true` на всех дверях

```
...  
  
- Id: 4  
  
  Title: "Дверь 4"  
  
  CommandToggleRegisterAddress: 12399  
  
  LockUnlockRegisterAddress: 12400  
  
  StateRegisterAddress: 12462  
  
  HasLocking: true  
  
  HasState: true  
  
...
```

Затем ниже в ветке `Alldoors`:

```
...  
  
AllDoors:  
  
  Title: "Все двери"  
  
  CommandOpenRegisterAddress: 12391  
  
  CommandCloseRegisterAddress: 12463  
  
  CommandLockUnlockRegisterAddress: 12392  
  
  # CommandOpenValue: 1;  
  
  # CommandCloseValue: 1;  
  
  # CommandLockValue: 1;  
  
  # CommandUnlockValue: 0;  
  
  CanOpen: true  
  
  CanClose: true  
  
  CanLock: true  
  
...
```

Импортировать файл *plc.config.default.yml*

```
# export ESC1_SERVER_API_CONF=/etc/mvs.esc1/Server.API.yml  
  
# sudo -u mvsservice --preserve-env=ESC1_SERVER_API_CONF  
/srv/MVS.ESC1.Server.API/ESC1.Server.Management.App ImportPlcVariablesMap  
/etc/mvs.esc1/plc.config.default.yml  
  
# systemctl restart ESC1.Server.API.service
```

## 6.2 Установка сервера на Ubuntu

Выполнить настройку из раздела Настройка Ubuntu: Подключение репозитория MVS

Error: Reference source not found

Установить mvs-esc1-server-api:

```
apt -y install mvs-esc1-server-api
```

Отредактировать `/etc/mvs.esc1/Server.API.yml` согласно примеру:

```
...

SettingsStorageOptions:

## long value. Maximum size (in bytes) that the datafile can grow to. Cannot be smaller 4
pages (32768 bytes).

# DatabaseLimitSize: 1048576

## See https://www.litedb.org/docs/

## e.g. 'Filename=/opt/settings.db'

ConnectionString: "/var/lib/mvs.esc1/settings.db"

EventsStorageOptions:

## See https://www.litedb.org/docs/

## e.g. 'Filename=/opt/events.db'

ConnectionString: "/var/lib/mvs.esc1/events.db"
```

```
TrendsStorageOptions:

## See https://www.litedb.org/docs/

## e.g. 'Filename=/opt/trends.db'

ConnectionString: "/var/lib/mvs.esc1/trends.db"

...
```

Установить *mvs-esc1-server-web*:

```
apt -y install mvs-esc1-server-web
```

Создать на основе */etc/nginx/conf.d/ESC1.Server.Web.conf.example* конфиг ниже.

```
vim /etc/nginx/conf.d/ESC1.Server.Web.conf
```

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    "" close;
}

# EDIT ESC1.Server.API port if changed

upstream mvs-esc1-server-api-upstream {
    server 127.0.0.1:3000;
}
```

```
server {  
  
    listen 80 default_server;  
  
    server_name _;  
  
    root /srv/MVS.ESC1.Server.Web;  
  
    if ($scheme != "https") {  
        rewrite ^ https://$host$uri permanent;  
    }  
}
```

```
server {  
  
    listen 80;  
  
    server_name host_ip;  
  
    root /srv/MVS.ESC1.Server.Web;  
  
    location / {  
  
        index index.html;
```



```
try_files $uri /index.html;

}

location /api {

    proxy_pass http://mvs-esc1-server-api-upstream;

    proxy_set_header    Host $host;

    proxy_http_version  1.1;

    proxy_set_header Upgrade $http_upgrade;

    proxy_set_header Connection $connection_upgrade;

}

location /hubs/scene {

    proxy_pass http://mvs-esc1-server-api-upstream;

    proxy_set_header    Host $host;

    proxy_http_version  1.1;

    proxy_set_header Upgrade $http_upgrade;

    proxy_set_header Connection $connection_upgrade;

}

}
```

Править `/etc/nginx/nginx.conf`:

```
user nginx;

worker_processes auto;

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

pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {

    include /etc/nginx/mime.types;

    default_type application/octet-stream;

    server_names_hash_bucket_size 64;
```

```
client_max_body_size 6400m;

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 buffer=16k flush=2m;

sendfile    on;

tcp_nopush  on;

tcp_nodelay on;

keepalive_timeout 65;

keepalive_requests 100;

server_tokens on;

gzip on;

include /etc/nginx/conf.d/*.conf;
```

```
}
```

Чтобы двери имели функцию блокировки:

Редактировать */etc/mvs.esc1/plc.config.default.yml*.

В ветке Doors выставить true на всех дверях:

```
...
```

```
- Id: 4
```

```
  Title: "Дверь 4"
```

```
  CommandToggleRegisterAddress: 12399
```

```
  LockUnlockRegisterAddress: 12400
```

```
  StateRegisterAddress: 12462
```

```
  HasLocking: true
```

```
  HasState: true
```

```
...
```

Затем ниже в ветке *Alldoors*:

```
...  
  
AllDoors:  
  
  Title: "Все двери"  
  
  CommandOpenRegisterAddress: 12391  
  
  CommandCloseRegisterAddress: 12463  
  
  CommandLockUnlockRegisterAddress: 12392  
  
#  CommandOpenValue: 1;  
  
#  CommandCloseValue: 1;  
  
#  CommandLockValue: 1;  
  
#  CommandUnlockValue: 0;  
  
  CanOpen: true  
  
  CanClose: true  
  
  CanLock: true  
  
...
```

Установить базу данных:

```
# export ESC1_SERVER_API_CONF=/etc/mvs.esc1/Server.API.yml
```

```
# sudo -u mvse1 --preserve-env=ESC1_SERVER_API_CONF
/srv/MVS.ESC1.Server.API/ESC1.Server.Management.App InitDatabase | grep New.password
> /etc/mvs.esc1/passwd

# sudo openssl genrsa -out /var/lib/mvs.esc1/esc1privkey.pem 2048

# sudo openssl rsa -in /var/lib/mvs.esc1/esc1privkey.pem -pubout -RSAPublicKey_out -out
/var/lib/mvs.esc1/esc1pubkey.pem

# cd /var/lib/mvs.esc1/

# chown mvse1:mvse1 *.pem

# sudo -u mvse1 --preserve-env=ESC1_SERVER_API_CONF
/srv/MVS.ESC1.Server.API/ESC1.Server.Management.App ImportRSAPrivateKey
/var/lib/mvs.esc1/esc1privkey.pem

# sudo -u mvse1 --preserve-env=ESC1_SERVER_API_CONF
/srv/MVS.ESC1.Server.API/ESC1.Server.Management.App ImportRSAPublicKey
/var/lib/mvs.esc1/esc1pubkey.pem

# sudo -u mvse1 --preserve-
env=ESC1_SERVER_API_CONF/srv/MVS.ESC1.Server.API/ESC1.Server.Management.App
ImportPlcVariablesMap /etc/mvs.esc1/plc.config.default.yml
```

```
nginx -s reload
```

```
systemctl enable ESC1.Server.API.service && systemctl restart ESC1.Server.API.service
```

Пароль для входа на Web-интерфейс инженерной системы в */etc/mvs.esc1/passwd*

### 6.3 - Установка клиента

```
apt update
```

```
apt install gpg xterm libva-glx2 libva-x11-2 i965-va-driver libgl1-mesa-glx
```

Выполнить пункт Подключение репозитория MVS

```
apt install chrony
```

Создание пользователя mvs:

```
adduser mvs  
passwd  
#mvs
```

Добавить пользователя mvs в группы sudo, audio, video:

```
usermod -a -G sudo,audio,video mvs
```

Установить пакет *mvs-esc1-terminal-app*:

```
apt install mvs-esc1-terminal-app
```

Редактировать файл */etc/mvs.esc1/ESC1.TerminalApp.yml*:

```
esc1:
  httpConnection:
    url: http://{{ ESC1_SERVER_IP }}
    retryDelay: 1000 # ms; must be >= 1000

  signalRConnection:
    url: http://{{ ESC1_SERVER_IP }}
    retryDelay: 3000 # ms; must be >= 1000

  controlPanel:
    id: "выдаётся через web-интерфейс master"
    token: "выдаётся через web-интерфейс master"
```

Проверить юнит запуска службы *ESC1.TerminalApp.service*:

```
Restart=always
RestartSec=15
```

Настройка автозапуска для пользователя *mvs*:

```
sudo su - mvs && cd ~
```

*vim .xsession*:



```
exec openbox-session
```

vim *.bash\_profile*:

```
[[ -z $DISPLAY && $XDG_VTNR -eq 1 ]] && exec startx -- -nocursor > /dev/null 2>&1
```

```
sudo mkdir -p /etc/systemd/system/getty@tty1.service.d/
```

vim */etc/systemd/system/getty@tty1.service.d/override.conf*:

```
[Service]
ExecStart=
ExecStart=-/sbin/agetty --noissue --autologin mvs %I $TERM
Type=idle
```

Дополнить */etc/xdg/openbox/rc.xml*:

```
<keybind key="C-A-t">
  <action name="Execute">
    <startupnotify>
      <enabled>true</enabled>
      <name>Starting xterm</name>
    </startupnotify>
    <command>xterm</command>
  </action>
</keybind>

<keybind key="C-A-BackSpace">
  <action name="Execute">
    <command>kill X</command>
  </action>
</keybind>
```

```
</action>  
</keybind>
```

Дополнить `/etc/xdg/openbox/autostart:`

```
xset s off  
xset s noblank  
xset -dpms  
  
amixer sset Master unmute  
amixer sset Headphone unmute  
amixer sset Front unmute  
  
amixer set Master 90%  
amixer set Headphone 90%  
amixer set Front 90%
```

```
reboot
```

Проверить состояние службы, не должно быть предупреждений.

## 7. Коридорный монитор (Intercom)

```
apt update && apt upgrade
```

Подключить репозитории MVS

Создание пользователя mvs:

```
adduser mvs  
passwd mvs  
#mvs
```

Установка ПО КИПУ Lyra

```
apt install xterm mvs-lyra3-or-intercom -y
```

Если во время установки будет ошибка «не найден файл» нужно выполнить:

```
cp /usr/share/mvs/default-config/config-LYRA3_OR_INTERCOME_CONFIG.yml  
/usr/share/mvs/default-config/config-LYRA3_OR_INTERCOM_CONFIG.yml
```

и снова запустить установку.

Создаем юнит запуска службы:

```
/lib/systemd/system/Lyra3.OR.Intercom.service
```

```
[Unit]  
Description=Lyra3.OR.Intercom  
Wants=graphical.target
```

```
After=graphical.target

[Service]
Type=simple
User=mvs
Group=mvs
PermissionsStartOnly=true
Environment="DISPLAY=:0"
Environment="XAUTHORITY=/home/mvs/.Xauthority"
Environment="LYRA3_OR_INTERCOM_CONFIG=/etc/mvs/config-
LYRA3_OR_INTERCOM_CONFIG.yml"
StandardOutput=file:/var/log/mvs/lyra_tap_intercom.log

Restart=always
RestartSec=3s
TimeoutSec=30

WorkingDirectory=/srv/Lyra3.OR.Intercom
ExecStart=/srv/Lyra3.OR.Intercom/lyra3-or-intercom

[Install]
WantedBy=graphical.target
```

Добавить в автозагрузку:

```
systemctl enable Lyra3.OR.Intercom.service
```

В файле конфигурации подставляем параметры подключения:

```
/etc/mvs/config-LYRA3_OR_INTERCOM_CONFIG.yml
```

```
---
```

rabbitConnection:

url: "amqp://rabbitmq:1oEXpuKjH4@{{ rabbitmq\_url }}:5672" # or  
"amqp://username:password@localhost:5672"

retryDelay: 1000 # ms

# Failed connection will be repeated after the lapse of defined time (ms).

# Cannot be less than 1000 ms

# http connection configuration

httpConnection:

url: "http://{{ OR\_IP }}:{{ OR\_Port }}"

retryDelay: 1000 # ms

# Failed connection will be repeated after the lapse of defined time (ms).

# Cannot be less than 1000 ms

hpConnection:

serverAddress: 'https://mvs.{{ HP\_domain }}.mvsservice.ru'

retryDelay: 3000

authorization:

hpKey: {{ HpKey }}

substreams:

sidebar: Low

screen:

layout1x1:

- High

layoutPiP:

- High

- Med

layout2x2:

- Med

- Med

- Med

- Med

ui:

hideCursor: true

# Surgery with no end time duration (minutes)

baseSurgeryDuration: 60

logger:

level: info # debug | info | warn | error

```
crashReporter:

enabled: false

# if crashReporter is disabled, the rest of its configuration is not validated

productName: 'Lyra3.OR.Intercom'

companyName: 'MVS'

submitURL: 'http://192.168.66.97:1127/post'
```

Настройка автозапуска для пользователя mvs:

```
sudo su - mvs
cd ~
```

vim *.xsession*

```
exec openbox-session
```

vim *.bash\_profile*

```
[[ -z $DISPLAY && $XDG_VTNR -eq 1 ]] && exec startx > /dev/null 2>&1
```

```
systemctl restart Lyra3.OR.Intercom.service
```

Дополнить */etc/xdg/openbox/rc.xml*:

```
<keybind key="C-A-t">
  <action name="Execute">
```

```
<startupnotify>
  <enabled>true</enabled>
  <name>Starting xterm</name>
</startupnotify>
<command>xterm</command>
</action>
</keybind>

<keybind key="C-A-BackSpace">
  <action name="Execute">
    <command>pkill X</command>
  </action>
</keybind>
```

Дополнить */etc/xdg/openbox/autostart:*

```
xset s off
xset s noblank
xset -dpms

amixer sset Master unmute
amixer sset Headphone unmute
amixer sset Front unmute

amixer set Master 90%
amixer set Headphone 90%
amixer set Front 90%
```

```
mkdir -p /etc/systemd/system/getty@tty1.service.d/
```

*/etc/systemd/system/getty@tty1.service.d/override.conf*



```
[Service]
```

```
ExecStart=
```

```
ExecStart=-/sbin/agetty --noissue --autologin mvs %I $TERM
```

```
Type=idle
```

```
reboot
```