

**УТВЕРЖДАЮ**

Директор направления  
развития новых бизнесов  
ООО «НЦИ»



\_\_\_\_\_ А.Ф. Кокин

« \_\_\_\_\_ » \_\_\_\_\_ 2020 г.

**УТВЕРЖДАЮ**

Заместитель руководителя  
Контрактной службы ФГБУ «НМИЦ АГП  
им. В.И. Кулакова» Минздрава России

\_\_\_\_\_ Л.А. Калачев

« \_\_\_\_\_ » \_\_\_\_\_ 2020 г.

**Специализированная вертикально-интегрированная  
медицинская информационная система «Акушерство и  
гинекология и «Неонатология» Министерства  
здравоохранения Российской Федерации**

**(ВИМИС «Акушерство и гинекология» и «Неонатология»)**

Контракт от 06.11.2020 № 2465/44-2020

**Руководство администратора**

**29431809.509000.002.И5**

на 48 листах

**Москва 2020**

## Содержание

<b>Перечень терминов и сокращений .....</b>	<b>4</b>
<b>1 Введение .....</b>	<b>6</b>
1.1 Область применения .....	6
1.2 Уровень подготовки пользователя.....	6
1.3 Перечень эксплуатационной документации, с которыми необходимо ознакомиться пользователю .....	7
<b>2 Подготовка инфраструктуры .....</b>	<b>8</b>
2.1 Аппаратные требования к Системе .....	8
2.1.1 Требования к серверному оборудованию.....	8
2.1.2 Требования к оборудованию АРМ пользователей Системы .....	8
2.1.3 Программные требования к Системе .....	9
2.2 Состав дистрибутива Системы .....	10
<b>3 Описание операций.....</b>	<b>12</b>
3.1 Настройка инфраструктуры.....	12
3.2 Запуск Системы.....	15
3.3 Остановка Системы .....	30
3.4 Перевод Системы из штатного режима в режим «Прием сообщений» .....	31
3.5 Перевод Системы в штатный режим .....	31
<b>4 Обслуживание Системы.....</b>	<b>32</b>
4.1 Резервное копирование БД.....	32
4.1.1 Внеплановое резервное копирование БД.....	34
4.2 Восстановление резервной копии БД.....	34
4.3 Логирование резервного копирования БД.....	35
4.4 Порядок снятия резервного копирования.....	35
4.5 Порядок диагностики, локализации и устранения проблем.....	36

4.5.1	<i>Диагностика, локализация и устранения проблем с доступностью серверов</i>	36
4.5.2	<i>Диагностика, локализация и устранения проблем дискового пространства</i>	36
4.5.2.1	<i>Закончилось свободное место на диске</i>	36
4.5.2.2	<i>Превышение лимитов на открытие файлов</i>	37
4.5.3	<i>Диагностика, локализация и устранения проблем загрузки процессора</i>	38
4.5.4	<i>Диагностика, локализация и устранения проблем СУБД</i>	38
4.6	Мониторинг блокировок СУБД	39
4.7	Процедура проверки функционирования ПК после восстановления резервной копии и перевода ПК в штатный режим функционирования	41
4.8	Оптимизация и настройки производительности	41
4.8.1	<i>Оптимизация производительности СУБД</i>	41
4.8.2	<i>Оптимизация работы nginx</i>	42
4.9	Порядок вывода ПК из эксплуатации	43
4.10	Аварийные ситуации	43
	<b>Приложение А</b>	<b>44</b>

## Перечень терминов и сокращений

Термин, сокращение	Определение
PostgreSQL	Свободная объектно-реляционная система управления базами данных
SOAP	Simple Object Access Protocol – протокол обмена структурированными сообщениями в распределённой вычислительной среде
TCP/IP	Набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет. Протоколы работают друг с другом в стеке (англ. stack, стопка) – это означает, что протокол, располагающийся на уровне выше, работает «поверх» нижнего, используя механизмы инкапсуляции. Например, протокол TCP работает поверх протокола IP
АРМ	Автоматизированное рабочее место
БД	База данных
ВИМИС «Акушерство и гинекология» и «Неонатология», Система	Специализированная вертикально-интегрированная медицинская информационная система по профилям «Акушерство и гинекология» и «Неонатология»
ВМ	Виртуальная машина
ГОСТ	Государственный стандарт
МИС	Медицинская информационная система
ОМП	Оказание медицинской помощи
ОС	Операционная система
ПК	Персональный компьютер
ПО	Программное обеспечение
ПСИ	Профильная справочная информация
СУБД	Система управления базами данных
СЭМД	Структурированный электронный медицинский документ
СЭМД beta-версии	Структурированный электронный медицинский документ beta-версии (ранее СМС (структурированные медицинские

<b>Термин, сокращение</b>	<b>Определение</b>
	сведения), с 29.09.2020 г. СЭМД beta-версии на основании протокола совещания от 29.09.2020 г. с Регламетной службой ФГБУ «ЦНИИОИЗ» Минздрава России)
ФЛК	Форматно-логический контроль
ФРМО	Федеральный регистр медицинских организаций
ФРМР	Федеральный регистр медицинских работников

## **1 Введение**

### **1.1 Область применения**

Настоящий документ является руководством администратора и предназначен для специалистов, ответственных за реализацию ИТ-решений в рамках Специализированной вертикально-интегрированной медицинской информационной системы по профилям «Акушерство и гинекология» и «Неонатология» (далее – ВИМИС «Акушерство и гинекология» и «Неонатология», Система), которым необходимо устанавливать, настраивать, обслуживать и обеспечивать работоспособность Системы. Руководство содержит основные требования по установке и системному сопровождению ВИМИС «Акушерство и гинекология» и «Неонатология», а также порядок конфигурирования базовых компонентов ПО и пути решения возможных проблем при эксплуатации Системы.

### **1.2 Уровень подготовки пользователя**

Администраторы ВИМИС «Акушерство и гинекология» и «Неонатология» должны быть квалифицированными специалистами в следующих областях:

- настройка и управление системами, основанными на многоуровневой клиент-серверной архитектуре в сетях на основе протокола TCP/IP;
- настройка и управление реляционными БД, СУБД PostgreSQL;
- знание технического английского языка на уровне понимания англоязычных инструкций по обслуживанию информационных систем;
- выполнение диагностики и предупреждение системных ошибок;

- анализ и решение инцидентов в рамках 2 и 3 линии сопровождения ВИМИС «Акушерство и гинекология» и «Неонатология» в рамках своей зоны ответственности;
- знание и опыт администрирования Linux систем: ОС Centos 7;
- опыт работы с Docker и системами оркестрации.

### **1.3 Перечень эксплуатационной документации, с которыми необходимо ознакомиться пользователю**

Пользователям Системы рекомендуется ознакомиться с настоящим документом.

## **2 Подготовка инфраструктуры**

### **2.1 Аппаратные требования к Системе**

#### **2.1.1 Требования к серверному оборудованию**

Для обеспечения высокой доступности и восстановления после аппаратных сбоев с минимальными незапланированными простоями для серверов БД и приложений должно быть предусмотрено использование минимум двух узлов кластера, поддерживающего режим высокой доступности.

Каждый узел кластера должен обладать следующими минимальными эксплуатационными характеристиками:

- поддержка нескольких процессоров;
- поддержка оперативной памяти с контролем четности большого объема;
- наличие межкомпонентного соединения с высокой пропускной способностью (не менее 1 Гб/сек);
- поддержка режима «горячей замены» для дисков;
- наличие избыточных источников питания с поддержкой режима «горячей замены» и независимыми шнурами питания;
- поддержка режима «горячей замены» вентиляторов;
- наличие подсистемы резервного копирования;
- размещение в стандартной 19-дюймовой промышленной стойке;
- наличие источника бесперебойного питания.

#### **2.1.2 Требования к оборудованию АРМ пользователей Системы**

АРМ пользователя должно иметь следующие характеристики:

- центральный процессор с тактовой частотой не ниже 2,4 ГГц на платформе Intel Core2Duo;



- оперативная память не менее 4 Гб;
- графическая карта с видеопамятью не менее 512 Мбайт;
- жесткий диск с памятью не менее 100 Гб;
- сетевая карта с пропускной способностью 1000 Мбит/сек;
- монитор с разрешением не менее 1920x1080 пикселей;
- акустическая система;
- сетевой фильтр;
- источник бесперебойного питания.

### 2.1.3 Программные требования к Системе

Перечень системного и программного обеспечения представлен в таблице (Таблица 1).

Таблица 1 – Перечень системного и программного обеспечения

№	ПО	Версия	Описание	Ссылка
1	Centos	7	Дистрибутив linux	<a href="https://www.centos.org/">https://www.centos.org/</a>
2	Docker	19.03	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации	<a href="https://www.docker.com/">https://www.docker.com/</a>
3	Docker-compose	1.27	Инструментом для определения и запуска контейнерных приложений	<a href="https://docs.docker.com/compose/">https://docs.docker.com/compose/</a>
4	Postgresql Server	11.8	Свободная объектно-реляционная система управления базами данных	<a href="https://www.postgresql.org/">https://www.postgresql.org/</a>
5	Cassandra	3.11	Распределенная система управления базами данных	<a href="https://cassandra.apache.org/">https://cassandra.apache.org/</a>
6	Kafka	2.5	Распределенный	<a href="https://kafka.apache.org/">https://kafka.apache.org/</a>

№	ПО	Версия	Описание	Ссылка
			программный брокер сообщений	
7	Zookeeper	3.6	Централизованная служба для поддержки информации о конфигурации	<a href="https://zookeeper.apache.org/">https://zookeeper.apache.org/</a>

## 2.2 Состав дистрибутива Системы

Система включает в себя следующий набор приложений (Таблица 2).

Таблица 2 – Набор приложений

№	Имя микросервера	Наименование подсистемы
1	vimis-dispatcher-flk	Сервис обработки этапа ФЛК, обработки СЭМД beta-версии
2	vimis-dispatcher-validation	Сервис обработки этапа XSD, обработки СЭМД beta-версии
3	vimis-gost-signature	Сервис ГОСТ подписи запросов
4	vimis-journal	Журналирование
5	vimis-method-back	Методология
6	vimis-netcore-tasks	Сервис интеграции с ФРМО/ФРМР, импорта лицензий, обогащения адресов
7	vimis-passport-back	Паспорт службы
8	vimis-patient-monitoring-back	Мониторинг пациентов
9	vimis-storage-back	Сервис обработки этапа сохранения СЭМД beta-версии в БД, обработки СЭМД beta-версии
10	vimis-storage-back-js	Хранилище данных
11	vimis-xslt	Сервис для xslt преобразований СЭМД/ СЭМД

№	Имя микросервера	Наименование подсистемы
		beta-версии
12	vimis-adapters	SOAP сервис приема СЭМД beta-версии, ОМП
13	vimis-adapters-callback-sender	Отправка обратных вызовов для МИС
14	vimis-adapters-frmo	SOAP сервис приема обратных вызовов ФРМО
15	vimis-admin	Администрирование
16	vimis-analytics-back	Аналитика
17	vimis-processing-patient-route	Сервис построения маршрута, контроля клинреков, и отклонений после приема СЭМД beta-версии
18	vimis-psi	ПСИ

### 3 Описание операций

#### 3.1 Настройка инфраструктуры

На всех ВМ от пользователя root (если пользователь указан неявно):

- обновите ОС;

```
yum upgrade
```

- установите дополнительные пакеты;

```
yum -y install net-tools mc vim
```

- настройте дату и время;

```
timedatectl set-timezone Europe/Moscow
```

```
yum install ntpdate ntp
```

```
chkconfig ntpd on
```

```
ntpdate pool.ntp.org
```

```
service ntpd start
```

- отключите selinux;

```
sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
```

```
reboot
```

- пропишите ВМ в /etc/hosts;

```
# host manager
192.168.129.2    prod-balancer-01
192.168.129.3    prod-balancer-02
# host worker
192.168.129.4    prod-app-01
192.168.129.5    prod-app-02
192.168.129.6    prod-app-03
```

```
192.168.129.7    prod-db-01
192.168.129.8    prod-db-11
192.168.129.9    prod-db-12
# host no swarm cluster
192.168.129.11   prod-backup-01
192.168.129.10   prod-share-01
```

– **добавьте пользователя admin;**

```
adduser admin
```

```
passwd admin
```

```
usermod -aG wheel
```

– **установите docker (от пользователя admin);**

```
sudo yum install -y yum-utils
```

```
sudo yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-
ce.repo
```

```
sudo yum install docker-ce docker-ce-cli containerd.io
```

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

```
sudo usermod -aG docker admin
```

– **установите docker-compose (от пользователя admin);**

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.27.4/d
ocker-compose-$(uname -s)-$(uname -m)" -o
/usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

```
sudo ln -s /usr/local/bin/docker-compose
/usr/bin/docker-compose
```

- **пройдите авторизацию в registry.bars.group (от пользователя admin);**

```
sudo docker login registry.bars.group
```

- **инициируйте docker swarm на хосте prod-balancer-01 (от пользователя admin);**

```
sudo docker swarm init --advertise-addr 192.168.129.2
```

- **откройте порты на хостах manager (от пользователя admin);**

```
sudo firewall-cmd --add-port=2376/tcp --permanent
```

```
sudo firewall-cmd --add-port=2377/tcp --permanent
```

```
sudo firewall-cmd --add-port=7946/tcp --permanent
```

```
sudo firewall-cmd --add-port=7946/udp --permanent
```

```
sudo firewall-cmd --add-port=4789/udp --permanent
```

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

- **подключите второй хост manager (от пользователя admin);**

```
sudo docker swarm join-token manager
```

```
# получаем token
```

```
# на хосте prod-balancer-02 выполняем команду
```

```
sudo docker swarm join --token <token> prod-balancer-01:2377
```

- **откройте порты на хостах worker (от пользователя admin);**

```
sudo firewall-cmd --add-port=2376/tcp --permanent
```

```
sudo firewall-cmd --add-port=7946/tcp --permanent
```

```
sudo firewall-cmd --add-port=7946/udp --permanent
```

```
sudo firewall-cmd --add-port=4789/udp --permanent
```

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

- перезагрузите сервис `docker` (от пользователя `admin`);

```
sudo systemctl restart docker
```

- отключите запуск контейнеров на хостах `manager` (от пользователя `admin`);

```
sudo docker node update --availability drain balancer-*
```

- на всех ВМ `prod-db-*` смонтируйте `vg_bigdisk` в `/var/lib/docker/volumes/` и добавьте монтирование при запуске системы в файл `</etc/fstab>`;

```
mount /dev/vg_bigdisk/data_volume
/var/lib/docker/volumes/
```

## 3.2 Запуск Системы

Выполните действия:

- создайте `compose` файл `docker-compose-app.yaml` и пропишите в нем параметры каждой подсистемы;

```
version: '3.8'
```

```
services:
```

```
  vimis-analytics-back:
```

```
    image: registry.bars.group/vimis/vimis-analytics-
back:latest
```

```
    environment:
```

```
      ANALYTIC_DATABASE_HOST: '192.168.233.230'
```

```
      ANALYTIC_DATABASE_PORT: '5432'
```

```
      ANALYTIC_DATABASE_NAME: 'analitic'
```

```
      ANALYTIC_DATABASE_USER: '***'
```

```
      ANALYTIC_DATABASE_PASSWORD: '***'
```

```
    deploy:
```

```
      placement:
```

```
        constraints:
```

```
          - node.role != manager
```

```

vimis-netcore-tasks:
  image: registry.bars.group/vimis/vimis-netcore-
tasks:latest
  environment:
    ApacheCassandra__Hosts: 'cassandra-node1:9042'
    ApacheKafka__BootstrapServers: 'kafka-node1:9092'
    PsiApiClient__BaseAddress: 'http://vimis-
psi:15555'
    SoapSignature__Type: 'Remote'
    SoapSignature__RemoteBaseAddress: 'http://vimis-
gost-signature'
    ConnectionStrings__DefaultConnection:
'Server=vimis-passport-service-
db;Port=5432;Database=passport_service;Username=app;Password=
***;Timeout=1024;'
    ConnectionStrings__PersDbConnection:
'Server=vimis-storage-
db;Port=5432;Database=vimis;Username=app;Password=***;Timeout
=1024;'
    VimisWeb__ServicesUrls: 'http://vimis-
adapters:5000;http://vimis-adapters-callback-
sender;http://vimis-adapters-frmo:5000;http://vimis-
dispatcher-validation:5000;http://vimis-dispatcher-
flk:5000;http://vimis-storage-back:5000;http://vimis-
processing-patient-route;http://vimis-gost-
signature;http://vimis-xslt;'
  deploy:
    placement:
      constraints:
        - node.role != manager

vimis-psi:
  image: registry.bars.group/vimis/vimis-psi:latest
  deploy:
    placement:
      constraints:
        - node.role != manager

vimis-adapters:
  image: registry.bars.group/vimis/vimis-
adapters:latest
  environment:
    ApacheCassandra__Hosts: 'cassandra-node1:9042'
    ApacheKafka__BootstrapServers: 'kafka-node1:9092'

```



```

        ConnectionStrings__DefaultConnection:
'Server=vimis-method-
db;Port=5432;Database=method;Username=app;Password=***;Timeou
t=1024;'
        SoapReceiver__SoapAddressLocation:
'https://vimis.egisz.rosminzdrav.ru/receiver'
        Clinrec__SoapAddressLocation:
'https://vimis.egisz.rosminzdrav.ru/clinrec'
        SoapSignature__Type: 'Remote'
        SoapSignature__RemoteBaseAddress: 'http://vimis-
gost-signature'
        SoapReceiver__DoCheckAccess: 'true'
        ConnectionStrings__SmsLogConnection:
'Server=vimis-storage-
db;Port=5432;Database=vimis;Username=app;Password=***;Timeout
=1024;'
        AdministrationApiClient__BaseAddress:
'http://vimis-admin:8004'
        deploy:
        placement:
        constraints:
        - node.role != manager

vimis-adapters-frmo:
  image: registry.bars.group/vimis/vimis-
adapters:latest
  environment:
    Logging__NLogConfigPath: 'nlog-frmo.config'
    ApacheKafka__BootstrapServers: 'kafka-nodel:9092'
    SoapSignature__Type: 'Remote'
    SoapSignature__RemoteBaseAddress: 'http://vimis-
gost-signature'
    Frmo__SoapAddressLocation:
'https://vimis.egisz.rosminzdrav.ru/frmo/receiver'
    ApacheCassandra__Hosts: 'cassandra-nodel:9042'
    ConnectionStrings__DefaultConnection:
'Server=vimis-method-
db;Port=5432;Database=method;Username=app;Password=***;Timeou
t=1024;'
    Clinrec__SoapAddressLocation:
'https://vimis.egisz.rosminzdrav.ru/clinrec'
    SoapReceiver__DoCheckAccess: 'false'
  deploy:
  placement:
  constraints:

```

```

- node.role != manager

vimis-adapters-callback-sender:
  image: registry.bars.group/vimis/vimis-adapters-
callback-sender:latest
  environment:
    ApacheCassandra__Hosts: 'cassandra-nodel:9042'
    ApacheKafka__BootstrapServers: 'kafka-nodel:9092'
    SoapSignature__Type: 'Remote'
    SoapSignature__RemoteBaseAddress: 'http://vimis-
gost-signature'
    ConnectionStrings__SmsLogConnection:
'Server=vimis-storage-
db;Port=5432;Database=vimis;Username=app;Password=***;Timeout
=1024;'
  deploy:
    placement:
      constraints:
        - node.role != manager

vimis-admin:
  image: registry.bars.group/vimis/vimis-admin:latest
  environment:
    DATABASE_HOST: 'vimis-admin-db'
    DATABASE: adm_master
    DATABASE_USER: app
    DATABASE_PORT: 5432
    DATABASE_PASSWORD: '***'
    DATABASE_MAX_CONNECTIONS: 20
  deploy:
    placement:
      constraints:
        - node.role != manager

vimis-app-onco:
  image: registry.bars.group/vimis/vimis-app-
onco:latest
  environment:
    EHR_HOST: 'http://vimis-storage-back-js:8001'
    PSI_HOST: 'http://vimis-psi:15555'
    ADMIN_HOST: 'http://vimis-admin:8004'
    JOURNAL_HOST: 'http://vimis-journal:8003'
    MONITORING_HOST: 'http://vimis-patient-monitoring-
back:8008'
    METHOD_HOST: 'http://vimis-method-back:8010'

```

```

PASSPORT_HOST: 'http://vimis-passport-back:8006'
KM_HOST: 'http://vimis-km-back:8009'
XSLT_HOST: 'http://vimis-xslt'
ANALYTICS_HOST: 'http://vimis-analytics-back:8012'
NETCORE_HOST: 'http://vimis-netcore-tasks'
PASSPORT_INTEGRATION_PROXY: 'https://vimis-
passport-integration:8000'
OLD_DATABASE_HOST: '192.168.233.42'
OLD_DATABASE_PORT: '5432'
OLD_DATABASE_NAME: 'nf_onko'
OLD_DATABASE_USER: 'nfadm'
OLD_DATABASE_PASSWORD: '***'
KAFKA_CLIENT_ID: 'Audit client'
KAFKA_BROKERS: 'kafka-nodel:9092'
ALPHABI_URL: 'http://egisz-dev.bars.group'
ALPHABI_URL_DASH: 'http://egisz-dev.bars.group'
ALPHABI_LOGIN: 'admin'
ALPHABI_PASSWORD: '***'
ALPHABI_DASH_OA_SMS_MON_ID: '48'
ALPHABI_DASH_OA_PATIENT_MON_ID: '49'
ALPHABI_DASH_OA_SERVICE_MON_ID: '50'
DEBUG_INCLUDE_TO_RESPONSE: 'true'
DEBUG_MSGS_EXTENDED_INFO: 'true'
deploy:
  placement:
    constraints:
      - node.role != manager

vimis-storage-back-js:
  image: registry.bars.group/vimis/vimis-storage-back-
js:latest
  environment:
    DATABASE_HOST: 'vimis-storage-db'
    DATABASE_PORT: '5432'
    DATABASE_NAME: 'vimis'
    APP_PASSWORD: '***'
    DEV_PASSWORD: '***'
    ADMIN_HOST: 'http://vimis-admin:8004'
    MONITORING_HOST: 'http://vimis-patient-monitoring-
back:8008'
    MIGRATE_DOMIGRATE: 'true'
    MIGRATE_CHECKTYPE: 'simple'
    DEBUG_NEED: 'true'
  deploy:
    placement:

```

```

        constraints:
            - node.role != manager

vimis-dispatcher-validation:
    image: registry.bars.group/vimis/vimis-dispatcher-
validation:latest
    environment:
        ApacheCassandra__Hosts: 'cassandra-nodel:9042'
        ApacheKafka__BootstrapServers: 'kafka-nodel:9092'
        ConnectionStrings__SmsLogConnection:
'Server=vimis-storage-
db;Port=5432;Database=vimis;Username=app;Password=***;Timeout
=1024;'
    deploy:
        placement:
            constraints:
                - node.role != manager

vimis-dispatcher-flk:
    image: registry.bars.group/vimis/vimis-dispatcher-
flk:latest
    environment:
        ApacheCassandra__Hosts: 'cassandra-nodel:9042'
        ApacheKafka__BootstrapServers: 'kafka-nodel:9092'
        PsiApiClient__BaseAddress: 'http://vimis-
psi:15555'
        PassportApiClient__BaseAddress: 'http://vimis-
passport-back:8006'
        PersApiClient__BaseAddress: 'http://vimis-storage-
back-js:8001'
        ConnectionStrings__SmsLogConnection:
'Server=vimis-storage-
db;Port=5432;Database=vimis;Username=app;Password=***;Timeout
=1024;'
    deploy:
        placement:
            constraints:
                - node.role != manager

vimis-storage-back:
    image: registry.bars.group/vimis/vimis-storage-
back:latest
    environment:
        ApacheCassandra__Hosts: 'cassandra-nodel:9042'
        ApacheKafka__BootstrapServers: 'kafka-nodel:9092'

```

```

        ConnectionStrings__DefaultConnection:
'Server=vimis-storage-
db;Port=5432;Database=vimis;Username=app;Password=***;Timeout
=1024;'
        PsiApiClient__BaseAddress: 'http://vimis-
psi:15555'
        PersApiClient__BaseAddress: 'http://vimis-storage-
back-js:8001'
        CheckingTest__XsdApiBaseAddress: 'http://vimis-
dispatcher-validation:5000'
        CheckingTest__FlkApiBaseAddress: 'http://vimis-
dispatcher-flk:5000'
        ConnectionStrings__SmsLogConnection:
'Server=vimis-storage-
db;Port=5432;Database=vimis;Username=app;Password=***;Timeout
=1024;'
        PassportApiClient__BaseAddress: 'http://vimis-
passport-back:8006'
        deploy:
        placement:
        constraints:
        - node.role != manager

vimis-journal:
  image: registry.bars.group/vimis/vimis-
journal:latest
  environment:
    PORT: 8003
    ApacheCassandra__Hosts: 'cassandra-node1:9042'
    ApacheCassandra__LocalDataCenter: 'datacenter1'
    ApacheKafka__BootstrapServers: 'kafka-node1:9092'
    ApacheKafka__ClientID: 'Audit client'
    ADMIN: 'http://vimis-admin:8004'
  deploy:
    placement:
    constraints:
    - node.role != manager

vimis-gost-signature:
  image: registry.bars.group/vimis/vimis-gost-
signature:latest
  deploy:
    replicas: 3
    placement:
    constraints:

```

```
- node.role != manager
```

```
vimis-km-back:
  image: registry.bars.group/vimis/vimis-km-
back:latest
  environment:
    DATABASE_HOST: 'vimis-control-db'
    DATABASE_PORT: '5432'
    DATABASE_NAME: 'control'
    APP_PASSWORD: '12qwaszx'
    DEV_PASSWORD: '1qazxsw2'
    PSI_HOST: 'http://vimis-psi:15555'
    ADMIN_HOST: 'http://vimis-admin:8004'
    PASSPORT_HOST: 'http://vimis-passport-back:8006'
    MONITORING_HOST: 'http://vimis-patient-monitoring-
back:8008'
    KM_HOST: 'http://vimis-km-back:8009'
    MIGRATE_DOMIGRATE: 'true'
    MIGRATE_CHECKTYPE: 'simple'
    DEBUG_NEED: 'true'
  deploy:
    placement:
      constraints:
        - node.role != manager
```

```
vimis-method-back:
  image: registry.bars.group/vimis/vimis-method-
back:latest
  environment:
    DATABASE_HOST: 'vimis-method-db'
    DATABASE_PORT: '5432'
    DATABASE_NAME: 'method'
    APP_PASSWORD: '***'
    DEV_PASSWORD: '***'
    MIGRATE_DOMIGRATE: 'true'
    MIGRATE_CHECKTYPE: 'simple'
    DEBUG_NEED: 'true'
  deploy:
    placement:
      constraints:
        - node.role != manager
```

```
vimis-passport-back:
  image: registry.bars.group/vimis/vimis-passport-
back:latest
```

```

environment:
  DATABASE_HOST: 'vimis-passport-service-db'
  DATABASE_PORT: '5432'
  DATABASE_NAME: 'passport_service'
  APP_PASSWORD: '***'
  DEV_PASSWORD: '***'
  ADMIN_HOST: 'http://vimis-admin:8004'
  EHR_HOST: 'http://vimis-storage-back-js:8001'
  MIGRATE_DOMIGRATE: 'true'
  MIGRATE_CHECKTYPE: 'simple'
  DEBUG_NEED: 'true'
deploy:
  placement:
    constraints:
      - node.role != manager

vimis-passport-integration:
  image: registry.bars.group/vimis/vimis-passport-
integration:latest
  deploy:
    placement:
      constraints:
        - node.role != manager

vimis-patient-monitoring-back:
  image: registry.bars.group/vimis/vimis-patient-
monitoring-back:latest
  deploy:
    placement:
      constraints:
        - node.role != manager

vimis-processing-patient-route:
  image: registry.bars.group/vimis/vimis-processing-
patient-route:latest
  environment:
    Check__KlinrekApiBaseAddress: 'http://vimis-
processing-klinrek'
    Check__DeviationApiBaseAddress: 'http://vimis-
processing-deviation'
    ApacheCassandra__Hosts: 'cassandra-nodel:9042'
    ApacheKafka__BootstrapServers: 'kafka-nodel:9092'
    MethodologyApiClient__BaseAddress: 'http://vimis-
method-back:8010/api'

```

```

        StorageRoutingApiClient__BaseAddress:
'http://vimis-storage-back-js:8001/api'
        PatientsMonitoringApiClient__BaseAddress:
'http://vimis-patient-monitoring-back:8008/api'
    deploy:
    placement:
        constraints:
            - node.role != manager

vimis-processing-klinrek:
    image: registry.bars.group/vimis/vimis-processing-
klinrek:latest
    environment:
        ApacheCassandra__Hosts: 'cassandra-nodel:9042'
        ApacheKafka__BootstrapServers: 'kafka-nodel:9092'
        MethodologyApiClient__BaseAddress: 'http://vimis-
method-back:8010/api'
        StorageRoutingApiClient__BaseAddress:
'http://vimis-storage-back-js:8001/api'
        PatientsMonitoringApiClient__BaseAddress:
'http://vimis-patient-monitoring-back:8008/api'
    deploy:
    placement:
        constraints:
            - node.role != manager

vimis-processing-deviation:
    image: registry.bars.group/vimis/vimis-processing-
deviation:latest
    environment:
        ApacheCassandra__Hosts: 'cassandra-nodel:9042'
        ApacheKafka__BootstrapServers: 'kafka-nodel:9092'
        MethodologyApiClient__BaseAddress: 'http://vimis-
method-back:8010/api'
        StorageRoutingApiClient__BaseAddress:
'http://vimis-storage-back-js:8001/api'
        PatientsMonitoringApiClient__BaseAddress:
'http://vimis-patient-monitoring-back:8008/api'
    deploy:
    placement:
        constraints:
            - node.role != manager

vimis-xslt:
    image: registry.bars.group/vimis/vimis-xslt:latest

```



```

environment:
  ApacheCassandra__Hosts: 'cassandra-node1:9042'
deploy:
  placement:
    constraints:
      - node.role != manager

```

- создайте compose файл docker-compose-kafka.yaml и пропишите в нем параметры для запуска kafka сервиса;

```
version: '3.8'
```

```
services:
```

```

zookeeper:
  image: bitnami/zookeeper:3.6.1
  environment:
    ALLOW_ANONYMOUS_LOGIN: 'yes'
  ports:
    - 2181:2181
  volumes:
    - zookeeper_data:/bitnami
  deploy:
    placement:
      constraints:
        - node.role != manager
        - node.id == jyc4l0s7f606taiamv9dr2gxx

```

```

kafka-node1:
  image: bitnami/kafka:2.5.0
  environment:
    KAFKA_CFG_ZOOKEEPER_CONNECT: zookeeper:2181
    KAFKA_LISTENERS:
CLIENT://:9092,EXTERNAL://:9094
    KAFKA_CFG_LISTENER_SECURITY_PROTOCOL_MAP:
CLIENT:PLAINTEXT,EXTERNAL:PLAINTEXT
    KAFKA_CFG_LISTENERS:
CLIENT://:9092,EXTERNAL://:9093
    KAFKA_CFG_ADVERTISED_LISTENERS:
CLIENT://kafka-node1:9092,EXTERNAL://kafka-node1:9093
    KAFKA_INTER_BROKER_LISTENER_NAME: CLIENT
    ALLOW_PLAINTEXT_LISTENER: 'yes'
    KAFKA_ZOOKEEPER_PROTOCOL: PLAINTEXT
  ports:
    - 9092:9092

```

```

    volumes:
      - kafka-node1-data:/bitnami
      -
./config/kafka/server.properties:/opt/bitnami/kafka/conf/server.properties
    deploy:
      placement:
      constraints:
        - node.role != manager
        - node.id == jyc4l0s7f606taiamv9dr2gxx

```

```

volumes:
  zookeeper_data:
    driver: local
  kafka-node1-data:
    driver: local

```

- создайте compose файл `docker-compose-cassandra.yaml` и пропишите в нем параметры для запуска `cassandra` сервиса;

```
version: '3.8'
```

```
services:
```

```

cassandra-node1:
  image: cassandra:3.11.6
  ports:
    - 9042:9042
  volumes:
    - cassandra-node1-data:/var/lib/cassandra
  deploy:
    placement:
      constraints:
        - node.role != manager
        - node.id == jyc4l0s7f606taiamv9dr2gxx

```

```

volumes:
  cassandra-node1-data:
    driver: local

```

- создайте compose файл `docker-compose-db.yaml` и пропишите в нем параметры для запуска `db` сервисов;

```
version: '3.8'
```

services:

```
vimis-admin-db:
  image: registry.bars.group/vimis/postgres:11.8
  volumes:
  - vimis-admin-db:/var/lib/postgresql/data:rw
  - type: tmpfs
    target: /dev/shm
    tmpfs:
      size: 268435456
  environment:
  POSTGRES_DB: 'adm_master'
  deploy:
  placement:
    constraints:
    - node.role != manager
    - node.id == ak5q4b06rvoietr9ecnlao1kw

vimis-method-db:
  image: registry.bars.group/vimis/postgres:11.8
  volumes:
  - vimis-method-db:/var/lib/postgresql/data:rw
  environment:
  POSTGRES_DB: 'method'
  deploy:
  placement:
    constraints:
    - node.role != manager
    - node.id == ak5q4b06rvoietr9ecnlao1kw

vimis-ref-book-db:
  image: registry.bars.group/vimis/postgres:11.8
  volumes:
  - vimis-ref-book-db:/var/lib/postgresql/data:rw
  environment:
  POSTGRES_DB: 'ref_book'
  deploy:
  placement:
    constraints:
    - node.role != manager
    - node.id == ak5q4b06rvoietr9ecnlao1kw

vimis-monitor-db:
  image: registry.bars.group/vimis/postgres:11.8
  volumes:
```

```

- vimis-monitor-db:/var/lib/postgresql/data:rw
environment:
POSTGRES_DB: 'monitor'
deploy:
placement:
  constraints:
  - node.role != manager
  - node.id == ak5q4b06rvoietr9ecnlao1kw

```

```

vimis-storage-db:
  image: registry.bars.group/vimis/postgres:11.8
  volumes:
  - vimis-storage-db:/var/lib/postgresql/data:rw
environment:
POSTGRES_DB: 'vimis'
deploy:
placement:
  constraints:
  - node.role != manager
  - node.id == ak5q4b06rvoietr9ecnlao1kw

```

```

vimis-passport-service-db:
  image: registry.bars.group/vimis/postgres:11.8
  volumes:
  - vimis-passport-service-
db:/var/lib/postgresql/data:rw
environment:
POSTGRES_DB: 'passport_service'
deploy:
placement:
  constraints:
  - node.role != manager
  - node.id == ak5q4b06rvoietr9ecnlao1kw

```

```

vimis-control-db:
  image: registry.bars.group/vimis/postgres:11.8
  volumes:
  - vimis-control-db:/var/lib/postgresql/data:rw
environment:
POSTGRES_DB: 'control'
deploy:
placement:
  constraints:
  - node.role != manager
  - node.id == ak5q4b06rvoietr9ecnlao1kw

```

```

vimis-am-db:
  image: registry.bars.group/vimis/postgres:11.8
  volumes:
    - vimis-am-db:/var/lib/postgresql/data:rw
    - type: tmpfs
      target: /dev/shm
      tmpfs:
        size: 268435456
  environment:
    POSTGRES_DB: 'bars_am'
  deploy:
    placement:
      constraints:
        - node.role != manager
        - node.id == ak5q4b06rvoietr9ecnlao1kw

```

```

volumes:
  vimis-admin-db:
    driver: local
  vimis-method-db:
    driver: local
  vimis-ref-book-db:
    driver: local
  vimis-monitor-db:
    driver: local
  vimis-storage-db:
    driver: local
  vimis-passport-service-db:
    driver: local
  vimis-control-db:
    driver: local
  vimis-am-db:
    driver: local

```

- создайте compose файл docker-compose-nginx.yaml и пропишите в нем параметры для запуска nginx сервисов;

```
version: '3.8'
```

```
services:
```

```

  nginx:
    image: nginx
    ports:

```

```

- 80:80
- 443:443
volumes:
-
./config/nginx/default.conf:/etc/nginx/conf.d/default.conf:rw
- ./config/nginx/ssl:/etc/nginx/ssl
command: [nginx-debug, '-g', 'daemon off;']
deploy:
placement:
  constraints:
  - node.id == 2xilu3gptqgzqnyv7677om7h6
  - node.id != jyc4l0s7f606taiamv9dr2gxh
  - node.id != ak5q4b06rvoietr9ecnlao1kw

```

– запустите сервисы db;

```
docker stack deploy --with-registry-auth -c "./docker-
compose-db.yml" production_stand
```

– запустите сервис cassandra;

```
docker stack deploy --with-registry-auth -c "./docker-
compose-cassandra.yml" production_stand
```

– запустите сервис kafka;

```
docker stack deploy --with-registry-auth -c "./docker-
compose-kafka.yml" production_stand
```

– запустите сервисы ВИМИС «Акушерство и гинекология» и «Неонатология»;

```
docker stack deploy --with-registry-auth -c "./docker-
compose-app.yml" production_stand
```

### 3.3 Остановка Системы

Выполните команду на хосте prod-balancer-01:

```
docker stack rm production_stand
```

### 3.4 Перевод Системы из штатного режима в режим «Прием сообщений»

Режим «Прием сообщений» необходим для обеспечения возможности обслуживания серверов БД (PostgreSQL) без остановки приема сообщений (СЭМД/ СЭМД beta-версии) от внешних информационных систем.

Для перевода Системы в режим «Прием сообщений» на хосте prod-balancer-01 отключите обработку СЭМД beta-версии:

```
docker service rm production_stand_vimis-dispatcher-validation
```

```
docker service rm production_stand_vimis-dispatcher-flk
```

```
docker service rm production_stand_vimis-storage-back
```

```
docker service rm production_stand_vimis-processing-patient-route
```

```
docker service rm production_stand_vimis-adapter-callback-sender
```

В режиме «Прием сообщений» должны работать следующие сервисы: cassandra, kafka, vimis-adapter-receiver, vimis-gost-signature, nginx.

### 3.5 Перевод Системы в штатный режим

На хосте prod-balancer-01 выполните действия:

- убедитесь в работе сервисов, необходимых для режима «Прием сообщений»;
- запустите сервисы ВИМИС «Акушерство и гинекология» и «Неонатология».

```
docker stack deploy --with-registry-auth -c "./docker-compose-app.yml" production_stand
```

## 4 Обслуживание Системы

### 4.1 Резервное копирование БД

Для настройки процедуры резервного копирования на сервере БД выполните следующие команды:

- создайте структуры директорий для резервных копий БД;

```
mkdir /pgbackup/db/{backup,log,scripts}
```

- создайте группу для резервного копирования, если ранее не создавалась;

```
sudo groupadd backup-database
```

- создайте пользователя для резервного копирования файлового хранилища;

```
sudo useradd -s /bin/bash -g backup -d /pgbackup/ -m  
backup-database
```

- задайте пароль для пользователя backup-database;

```
sudo useradd -s /bin/bash -g backup -d /pgbackup/ -m  
backup-database
```

- задайте права на директорию;

```
sudo chown -R backup-database:backup /pgbackup/
```

- создайте скрипт для снятия резервной копии БД  
/pgbackup/db/scripts/pgbackup\_db.sh со следующим содержимым:

```
#!/bin/bash  
DBASE="dbname" #название базы данных передается во  
время вызова скрипта  
DB_OWNER="backup" #пользователь, под которым происходит  
подключение
```



```

DB_PASS="backup" #пароль
DB_HOST="pg-0" #адрес сервера СУБД

BACKUP_DIR="/pgbackup/db/backup" #путь к хранилищу
резервных копий
LOG_DIR="/pgbackup/db/log" # путь до каталога с логами
KEEP_BACKUP="120" #количество хранимых копий (4 месяца)
THREADS=3 #количество потоков
BIN='/usr/pgsql-9.6/bin' #путь к каталогу с актуальной
версией pg_dump
DATE=$(date +%F-%k-%M)
find $BACKUP_DIR -name "*.backup" -mtime +${KEEP_BACKUP}
-delete > ${LOG_DIR}/${DATE}.log 2>&1

PGPASSWORD=${DB_PASS} ${BIN}/pg_dump -U ${DB_OWNER} -h
${DB_HOST} -Fd -j${THREADS} -v --blobs $DBASE --file
${BACKUP_DIR}/${DATE}_${DBASE}.pgsql.backup >>
${LOG_DIR}/${DATE}.log 2>&1

DUMP_SIZE=$(du -sh
${BACKUP_DIR}/${DATE}_${DBASE}.pgsql.backup | awk '{print $1}
')
COMPLETED=$(ls -lt ${BACKUP_DIR}/ | awk 'NR==2{print
$6,$7,$8} ')
cat > /pgbackup/db/log/dump.log<< EOF
# Название БД:
$DBASE
# Размер резервной копии:
$DUMP_SIZE
# Снятие завершено:
$COMPLETED"
# Лог бэкапа:
${LOG_DIR}/${DATE}.log

```

- для обеспечения возможности запускать скрипт назначьте права доступа на файл (скрипт) /pgbackup/db/scripts/pgbackup\_db.sh следующей командой:

```
sudo chmod +x /pgbackup/db/scripts/pgbackup_db.sh
```

- настройте запуск скрипта резервного копирования БД по расписанию;

```
sudo crontab -u backup-database -e
```

– вставьте строки:

```
0 22 * * * /pgbackup/db/scripts/pgbackup_db.sh
```

#### 4.1.1 Внеплановое резервное копирование БД

Выполните команду для снятия бэкапа БД:

```
sudo -u backup-database /pgbackup/scripts/pgbackup_db.sh
```

#### 4.2 Восстановление резервной копии БД

На сервере БД создайте БД, в которую будет происходить восстановление бэкапа:

```
sudo -u postgres createdb -O user dbname
```

где:

- dbname – пример названия БД;
- user — имя пользователя, под которым будет происходить восстановление БД.

Восстановите бэкап БД мониторинга, данное действие производится с VM резервного копирования:

```
pg_restore -U user -h ip_address -Fd -v -j3 -d dir_for_dump >>/pgbackup/restore.log 2>&1
```

где:

- user – имя пользователя, под которым производится восстановление БД;
- ip\_address – IP-адрес сервера СУБД;
- dir\_for\_dump – директория резервной копии;
- /pgbackup/restore.log – лог восстановления БД.

Зафиксируйте возникшую проблему, передав лог для обработки, если при проверке данного файла лога обнаружены ошибки вида:

- Failed;
- Error;
- Fatal;
- Warning.

### 4.3 Логи́рование резервного копирования БД

Логи́рование резервного копирования БД мониторинга осуществляется на ВМ резервного копирования в файл `/pgbackup/${DBNAME}/log/${DATE}.log`, где:

- `{DATE}` – дата снятия бэкапа в формате ГОД-МЕСЯЦ-ДЕНЬ-ЧАС-МИНУТА;
- `{DBNAME}` – имя БД, для которой производилось резервное копирование.

Зафиксируйте возникшую проблему, передав лог для обработки, если при проверке данного файла лога обнаружены ошибки вида:

- Failed;
- Error;
- Fatal;
- Warning.

### 4.4 Порядок снятия резервного копирования

Очередность резервного копирования должна быть организована по подсистемам, использующим СУБД:

- ПС01 Хранение;
- ПС08 Мониторинг пациентов;
- ПС10 Методология;
- ПС06 Паспорт службы;
- ПС11 ПСИ;

– PC04 Администрирование.

Имена БД приведены в приложении (Приложение А).

## **4.5 Порядок диагностики, локализации и устранения проблем**

### **4.5.1 Диагностика, локализация и устранения проблем с доступностью серверов**

В случае диагностирования недоступности сервера ПК более пяти минут зафиксируйте возникшую проблему, после чего выполните перезагрузку VM.

Если перезагрузка VM невозможна, либо после перезагрузки диагностируется недоступность сервера, войдите в консоль администрирования виртуализации и выясните причины недоступности VM.

### **4.5.2 Диагностика, локализация и устранения проблем дискового пространства**

Могут быть диагностированы следующие проблемы дискового пространства ПК:

- закончилось свободное место на диске;
- превышены лимиты на открытие файлов;
- другие ошибки.

#### **4.5.2.1 Закончилось свободное место на диске**

При обнаружении ошибки вида «No space left on device» на сервере, у которого закончилось свободное место на диске, для диагностирования проблемных папок выполните команду:

```
df -h
```

По команде будет выдан перечень папок с указанием занимаемого места на диске в подобном виде:

```
Filesystem Size Used Avail Use% Mounted on
/dev/mapper/vg_boportalem-lv_root 230G 219G 16M 100% /
```

Зафиксируйте возникшую ошибку и выберите один из вариантов решения:

- чистка ненужных логов ПК, установленного на сервере;
- увеличение раздела, на котором закончилось место.

#### 4.5.2.2 Превышение лимитов на открытие файлов

При обнаружении ошибки вида «Too many open files» выполните следующие действия:

- зафиксируйте возникшую ошибку;
- на сервере, на котором возникла проблема, в `/etc/sysctl.conf` добавьте строку (значение параметра должно выбираться исходя из текущих настроек ПК):

```
fs.file-max = 65536
```

- выполните команду:

```
sysctl -p
```

- в файл `/etc/security/limits.conf` добавьте (значения должны быть ниже `fs.file-max`):

```
* soft nofile 32768
* hard nofile 32768
```

- выполните перезапуск серверов, при работе которых возникала ошибка.

### 4.5.3 Диагностика, локализация и устранения проблем загрузки процессора

При длительной (более 1 часа) высокой загрузке процессора (более 95%) выполните следующие действия:

- зафиксируйте возникшую ошибку;
- проведите анализ перечня процессов на предмет выявления проблемного процесса;
- в зависимости от процесса, вызывающего критическую загрузку процессора, выполните либо завершение процесса, либо детальный анализ логов для выявления причин возникновения проблемы с последующим формированием предложений по ее устранению.

### 4.5.4 Диагностика, локализация и устранения проблем СУБД

Автоматическая «очистка мусора» БД.

В ходе функционирования ПК в СУБД могут накапливаться системные данные, оставшиеся от незавершенных по различным причинам операций. Они способны влиять на производительность ПК.

Для удаления этих данных необходимо еженедельно или при снижении производительности СУБД выполнять автоматическую «очистку мусора» встроенными средствами СУБД PostgreSQL.

Для этого на сервере БД выполните следующую команду:

```
vacuumdb -h host -p port -U user -d database -f -v -z
```

где:

- host – адрес/имя хоста;
- port – порт, на котором работает СУБД;
- user – имя пользователя;
- database – имя БД.

Для автоматизации процесса возможно использование cron. Для настройки периодического выполнения команды:

- откройте на редактирование файл заданий:

```
crontab -e -u postgres
```

- вставьте правило периодического выполнения в следующем формате (пример):

```
* * * * * vacuumdb -h host -p port -U user -d database -f -v -z
```

где:

```
* * * * *
- - - - -
| | | | |
| | | | ----- день недели (0 - 7) (Воскресенье =0 или
=7) ;
| | | ----- месяц (1 - 12) ;
| | ----- день (1 - 31) ;
| ----- час (0 - 23) ;
----- минута (0 - 59) .
```

#### 4.6 Мониторинг блокировок СУБД

В ходе функционирования ПК на СУБД могут возникать критические блокировки данных, существенно влияющие на производительность ПК.

В случае снижения производительности ПК и периодически (еженедельно) осуществляйте мониторинг блокировок СУБД следующим образом:

- подключитесь к СУБД:

```
psql -h host -p port -U user -d database
```

- host – укажите адрес/имя хоста, к которому производится подключение;
- port – порт, на котором работает СУБД;

- user – имя пользователя;
- database – имя БД, к которой производится подключение.
- просмотрите блокировки в СУБД в текущий момент:

```

SELECT blocked_locks.pid AS
blocked_pid,blocked_activity.username AS
blocked_user,blocking_locks.pid AS
blocking_pid,blocking_activity.username AS
blocking_user,blocked_activity.query AS blocked_statement,
blocking_activity.query AS
current_statement_in_blocking_process[1]FROM
pg_catalog.pg_locks blocked_locks JOIN
pg_catalog.pg_stat_activity blocked_activity ON
blocked_activity.pid = blocked_locks.pid JOIN
pg_catalog.pg_locks blocking_locks ON blocking_locks.locktype
= blocked_locks.locktype AND blocking_locks.DATABASE IS NOT
DISTINCT FROM blocked_locks.DATABASE AND
blocking_locks.relation IS NOT DISTINCT FROM
blocked_locks.relation AND blocking_locks.page IS NOT
DISTINCT FROM blocked_locks.page AND blocking_locks.tuple IS
NOT DISTINCT FROM blocked_locks.tuple AND
blocking_locks.virtualxid IS NOT DISTINCT FROM
blocked_locks.virtualxid AND blocking_locks.transactionid IS
NOT DISTINCT FROM blocked_locks.transactionid AND
blocking_locks.classid IS NOT DISTINCT FROM
blocked_locks.classid AND blocking_locks.objid IS NOT
DISTINCT FROM blocked_locks.objid AND blocking_locks.objsubid
IS NOT DISTINCT FROM blocked_locks.objsubid AND
blocking_locks.pid != blocked_locks.pidJOIN
pg_catalog.pg_stat_activity blocking_activity ON
blocking_activity.pid = blocking_locks.pid WHERE NOT
blocked_locks.GRANTED

```

В случае обнаружения критических блокировок обратитесь к разработчикам ПК.



## **4.7 Процедура проверки функционирования ПК после восстановления резервной копии и перевода ПК в штатный режим функционирования**

Для проверки функционирования ПК:

- откройте главную страницу ПК;
- войдите в личный кабинет пользователя;
- при наличии доступа перейдите в административную консоль.

## **4.8 Оптимизация и настройки производительности**

### **4.8.1 Оптимизация производительности СУБД**

В ходе эксплуатации СУБД PostgreSQL с ростом объемов данных, а также в результате интенсивных операций удаления и обновления данных могут наблюдаться снижение производительности, замедление выполнения запросов, рост загрузки процессора и повышенная утилизация подсистемы дискового ввода-вывода. Универсального решения по повышению производительности не существует, однако есть набор методов, сочетание которых позволит избежать негативных последствий:

- увеличение объема оперативной памяти под разделяемый буфер (параметр конфигурации `shared_buffers`). Рекомендуемый объем – 40% от объема оперативной памяти, но не более 8 GB. Приводит к снижению интенсивности дискового ввода-вывода;
- выставление значения `effective_cache_size` в 50% от объема оперативной памяти. Это позволит оптимизатору запросов точнее определить объем файлового кэша и выбрать оптимальный план выполнения запроса;
- увеличение точности собираемой статистической информации. Увеличьте параметр `default_statistics_target` до значения 100–1000.

Это должно положительно сказаться на построении оптимальных планов выполнения запросов;

- если наблюдается снижение производительности запросов, производящих объемные выборки данных, сортировки и группировку данных, – постепенное увеличение параметра `work_mem`, с постоянным контролем над поведением СУБД. Более точное значение позволит определить анализ плана выполнения запроса до и после изменения параметра;
- проведение операции очистки старых версий записей.

#### 4.8.2 Оптимизация работы nginx

Настройка рабочих процессов:

- при большом количестве сетевых соединений ресурсов 1 ядра процессора может быть недостаточно, и потребуются увеличение процессов – обработчиков соединений. Увеличьте значение `worker_processes` до числа физических ядер процессора;
- если после увеличения предыдущего значения наблюдаются проблемы на стороне пользователей, и при этом утилизация процессора не полная, увеличьте параметр `worker_connections` до значений 1024–2048 соединения на рабочий процесс.

Для приема множества соединений с целью снижения нагрузки на процессор и увеличения максимального числа одновременно обрабатываемых соединений переключите директиву `multi_accept` в значение «on».

Включите динамический парсер регулярных выражений, если в конфигурации используются регулярные выражения. Переключение параметра `pcre_jit` в значение «on» позволит ускорить сопоставление входящих запросов правилам конфигурации.

#### **4.9 Порядок вывода ПК из эксплуатации**

После окончания срока эксплуатации ПК (5 лет с момента ввода в эксплуатацию, если иное не определено дополнительно) выполните следующие процедуры:

- сформируйте копию БД ПК;
- загрузите сформированные копии БД ПК, документации ПК на носители информации для хранения;
- передайте сформированный согласно п. 2 пакет должностному лицу, ответственному за дальнейшее хранение информации. Должностное лицо, ответственное за дальнейшее хранение информации, должно определяться приказом по организации, являющейся оператором ПК.

#### **4.10 Аварийные ситуации**

Перечень регламентируемых аварийных ситуаций:

- отключение электроэнергии;
- сбой или выход из строя технических средств, на которых функционирует программное обеспечение Системы (за исключением физического разрушения);
- разрыв связи и (или) отказ телекоммуникационного оборудования;
- сбой в работе общесистемного или прикладного программного обеспечения Системы.

После получения сообщения об ошибке необходимо вернуться на предыдущую страницу (с помощью кнопки «Назад» веб-браузера) и попытаться повторить действие, которое привело к ошибке.

## Приложение А

### Типовой состав серверов ВИМИС «Акушерство и гинекология» и «Неонатология»

Типовой состав серверов ВИМИС «Акушерство и гинекология» и «Неонатология» приведен в таблице (Таблица А.1).

Таблица А.1 – Типовой состав серверов ВИМИС «Акушерство и гинекология» и «Неонатология»

Подсистема ВИМИС «Акушерство и гинекология» и «Неонатология»	№	Имя виртуальных машин	Роль	Имена исполняемых модулей, службы, сервисы	База данных
<b>Группа серверов БД и очереди Kafka</b>					
ПС01 Хранилище данных	1	vimis-storage-db-master	Мастер шардов	postgres	vimis
	2	vimis-storage-db-shard01	Шард 01	postgres	
	3	vimis-storage-db-shard02	Шард 02	postgres	
	4	vimis-storage-db-shard03	Шард 03	postgres	
	5	vimis-storage-db-shard04	Шард 04	postgres	
ПС02 Обработка, ПС03 Журналирование, ПС05 Адаптеры взаимодействия	6	vimis-cassandra-1	Кассандра нода 1	cassandra	vimis_exchange vimis_journal
	7	vimis-cassandra-2	Кассандра нода 2	cassandra	
	8	vimis-cassandra-3	Кассандра нода 3	cassandra	
	9	vimis-kafka-2	Кафка	kafka,	

Подсистема ВИМИС «Акушерство и гинекология» и «Неонатология»	№	Имя виртуальных машин	Роль	Имена исполняемых модулей, службы, сервисы	База данных
				zookeeper	
	10	vimis-kafka-2	Кафка	kafka, zookeeper	
ПС04 Администрирован ие	11	vimis-admin-db-1	База данных мастер	postgres	adm_master
	12	vimis-admin-db-2	База данных реплика	postgres	
ПС06 Паспорт Службы	13	vimis-passport- service-db-master	База данных	postgres	passport_servic e
	14	vimis-passport- service-db-slave	База данных	postgres	
ПС08 Мониторинг пациентов	15	vimis-patient-mon- db-1	База данных мастер	postgres	monitor
	16	vimis-patient-mon- db-2	База данных реплика	postgres	
ПС10 Методология	18	vimis-method-db	База данных	postgres	method
ПС11 ПСИ	13	vimis-psi-db-master	База данных	postgres	ref_book
	14	vimis-psi-db-slave	База данных	postgres	
ПС12 Аналитика	19	vimis-analytics-db- master	База данных мастер	postgres	analitic analitic_back
	20	vimis-analytics-db- shard01	База данных читающая реплика	postgres	
	21	vimis-analytics-db- shard02	База данных читающая реплика	postgres	

Подсистема ВИМИС «Акушерство и гинекология» и «Неонатология»	№	Имя виртуальных машин	Роль	Имена исполняемых модулей, службы, сервисы	База данных
	22	vimis-analytics-db-shard03	База данных читающая реплика	postgres	
<b>Группа серверов под управлением Docker Swarm</b>					
Балансировщики	23	vimis-balance-swarm-1		nginx, manager docker swarm	
	24	vimis-balance-swarm-2		nginx, manager docker swarm	
	25	vimis-balance-swarm-3		nginx, manager docker swarm	
Группа приложений ОБМЕН	26	vimis-exchange-01	Исполнение экземпляров контейнеров приложений	vimis-adapters- callback- receiver	
	27	vimis-exchange-02		vimis-adapters- callback-sender vimis-adapter vimis-adapter- frmo vimis-gost- signature	
Группа приложений Обработка	28	vimis-processing-1	Исполнение экземпляров контейнеров приложений	vimis- dispatcher-flk vimis- dispatcher- validation vimis-netcore- tasks vimis- processing- klinrek vimis- processing- patient-route	
	29	vimis-processing-2			

Подсистема ВИМИС «Акушерство и гинекология» и «Неонатология»	№	Имя виртуальных машин	Роль	Имена исполняемых модулей, службы, сервисы	База данных
				vimis- processing- deviation  vimis-storage- back  vimis-xslt	
Группа приложений БЭК	30	vimis-back-1	Исполнение экземпляров контейнеров приложений	vimis-admin	
	31	vimis-back-2		vimis-journal	
	32	vimis-back-3		vimis-km-back  vimis-method- back  vimis-passport- back  vimis-patient- monitoring-back  vimis-psi  vimis-storage- back-js	
Группа приложений ФРОНТ	33	vimis-desktop-app- 1	Исполнение экземпляров контейнеров приложений	vimis-app-onco	
	34	vimis-desktop-app- 2			
<b>Мониторинг здоровья системы</b>					
ELK стек	35	vimis-elk	Исполнение экземпляров контейнеров приложений	Logstash  Kibana  Elastic Search	
<b>Серверы хранения архивов</b>					
Бэкапы БД	36	vimis-backup-db	Сервер бэкапов баз данных		

