

УТВЕРЖДЕНО

Генеральный директор

ООО «К-СКАЙ»

Новицкий Р.Э.

03.04.2020 г.

**ПЛАТФОРМА ПРОГНОЗНОЙ АНАЛИТИКИ И
УПРАВЛЕНИЯ РИСКАМИ В ЗДРАВООХРАНЕНИИ
НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ
«WEBIOMED»**

РУКОВОДСТВО ПО УСТАНОВКЕ

И АДМИНИСТРИРОВАНИЮ

40910226.943119.005

Ине. № подл.	Подпись и дата	Взам. инв. №	Ине. № дубл.	Подпись и дата

Содержание

1	Введение	4
2	Термины и определения	5
3	Общие сведения по установке платформы.....	6
4	Планирование установки	7
	4.1 Общие сведения	7
	4.2 Варианты установки	8
	4.2.1 Требования к среде исполнения (выбор оркестратора).....	8
	4.2.2 Требования к отказоустойчивости.....	9
	4.2.3 Требования к объему обрабатываемых данных	10
	4.2.4 Типы установок.....	11
	4.3 Технические требования	12
	4.3.1 Требования к оборудованию	12
	4.3.2 Требования к программному обеспечению	13
	4.3.3 Другие требования.....	13
	4.4 Планирование масштабирования	15
	4.4.1 Особенности хранения данных	15
5	Подготовка к установке.....	17
	5.1 Установка необходимого ПО.....	17
	5.1.1 Установка Docker.....	17
	5.1.2 Установка Docker-compose	18
	5.1.3 Настройка серверов	18
	5.2 Настройка кластера Docker Swarm.....	19
	5.2.1 Общие сведения	19
	5.2.2 Создание master-узла	20
	5.2.3 Подключение worker-узлов к кластеру	20
	5.3 Подготовка дистрибутива платформы.....	21
6	Установка	22
	6.1 Общее описание	22

<i>Подпись и дата</i>		
<i>Име. № дубл.</i>		
<i>Взам. име. №</i>		
<i>Подпись и дата</i>		
<i>Име. № подл.</i>		

						<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		2

6.2	Установка локального реестра образов	22
6.2.1	Установка.....	22
6.2.2	Настройки для использования Registry	23
6.3	Установка платформы	24
6.4	Установка инструментов мониторинга	26
7	Настройка	27
7.1	Настройка системных параметров платформы.....	27
7.2	Настройка резервного копирования.....	27
8	Обслуживание	29
8.1	Управление сервисами платформы.....	29
8.2	Обновление	29
8.3	Масштабирование	30
8.4	Резервное копирование	32
8.4.1	Ручное создание резервных копий.....	32
8.4.2	Восстановление	32
9	Удаление платформы	34
10	Приложения.....	35
10.1	Сервисы платформы	35
10.2	Параметры установочных скриптов	40
10.3	Метки узлов кластера Docker Swarm.....	42
10.4	Описание параметров файлов настроек	45
10.4.1	Описание параметров системных настроек	45
10.4.2	Описание параметров приложения Dataset.....	47
10.4.3	Описание параметров приложения DHRA	47
10.4.4	Описание параметров приложения NLP	47
10.4.5	Описание параметров приложения Symptom-checker ...	48
10.4.6	Описание параметров приложения PatientID	48
10.4.7	Описание параметров приложения Gate	49
10.4.8	Описание параметров приложения DMA	49

Ине. № подл.	Подпись и дата					Лист
Ине. № дубл.	Подпись и дата					3
Взам. инв. №	Подпись и дата					
		Изм.	Лист	№ докум.	Подпись	Дата

1 Введение

В документе представлена информация, необходимая для установки и администрирования платформы прогнозной аналитики и управления рисками в здравоохранении на основе машинного обучения «Webiomed» (далее – «Платформа» или «Webiomed»).

Инев. № подл.	Подпись и дата	Взам. инв. №	Инев. № дубл.	Подпись и дата		Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

2 Термины и определения

Платформа – совокупность программного обеспечения, реализующего платформу прогнозной аналитики и управления рисками в здравоохранении на основе машинного обучения «Webiomed».

Образ (образ Docker) – пакет со всеми зависимостями и сведениями, необходимыми для создания контейнера. Образ содержит в себе базовый образ операционной системы, код приложения, и необходимые библиотеки для запуска приложения.

Контейнер (контейнер Docker) – экземпляр образа Docker, запущенный в среде исполнения Docker.

Сервис (сервис Docker) – один или более контейнеров, использующих один и тот же образ и запущенных с одинаковыми параметрами. Средствам Docker осуществляется балансировка нагрузки между контейнерами сервиса и поддержание необходимого количества запущенных контейнеров.

Реестр образов – централизованное хранилище образов Docker.

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата						
Изм.	Лист	№ докум.	Подпись	Дата						Лист
										5

3 Общие сведения по установке платформы

Платформа «Webiomed» разворачивается с использованием технологии контейнеризации приложений на базе программного обеспечения Docker (<https://www.docker.com/>, лицензия: Apache License 2.0), что позволяет создавать гибко масштабируемый и отказоустойчивый сервис. Docker берет на себя все операции по организации и обеспечению жизненного цикла контейнеров приложений: запуск, остановка, перезапуск при ошибках в работе, подключение к сети, масштабирование (увеличение и уменьшение количества контейнеров), т.е. выполняет функции так называемого оркестратора.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата					Лист
									6
					Изм.	Лист	№ докум.	Подпись	Дата

4 Планирование установки

4.1 Общие сведения

На основании данных о предполагаемых условиях функционирования системы, предъявляемых требований, технических характеристик устанавливаемой версии Webiomed, имеющихся вычислительных ресурсах должны быть определены:

- Вариант установки платформы.
- Требования по масштабированию Webiomed.
- Количественно-качественный состав установки: количество серверов, их характеристики и назначение.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата					Лист
									7
Изм.	Лист	№ докум.	Подпись	Дата					

4.2 Варианты установки

Вначале требуется определить условия работы платформы и предполагаемый характер ее использования. На основании этих данных можно будет определить какие вычислительные ресурсы потребуются выделить для установки Webiomed и как их использовать.

4.2.1 Требования к среде исполнения (выбор оркестратора)

Установка компонентов Webiomed производится только в виде контейнеров Docker. Для управления контейнерами в настоящее время поддерживаются и могут применяться следующие системы:

Docker-compose (<https://github.com/docker/compose>) — утилита для запуска многоконтейнерных приложений;

Docker Swarm (или Docker Swarm mode) (<https://docs.docker.com/engine/swarm/>) — Docker в режиме работы оркестратора.

С использованием Docker-compose можно быстрее развернуть платформу, т.к. требуется меньше операций по установке и настройке, но при этом система сможет работать только на одном сервере. При использовании Docker Swarm обычно создается и настраивается кластер из нескольких серверов, что позволяет создать масштабируемую и отказоустойчивую систему.

Исходя из возможностей инструментов управления контейнерами, рекомендуется их применять следующим образом:

Docker-compose - для тестирования или демонстраций возможностей Webiomed с установкой на персональный компьютер или тестовый сервер.

Docker Swarm - для промышленного использования Webiomed с установкой на группу серверов или один сервер, но с возможностью расширения их количества.

Ине. № подл.	Подпись и дата
Взам. инв. №	Ине. № дубл.
Подпись и дата	

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					8

4.2.2 Требования к отказоустойчивости

Под отказоустойчивостью здесь понимается способность платформы полноценно продолжать работать при выходе из строя каких-либо ее компонентов.

Обеспечение отказоустойчивости не гарантирует непрерывность работы всех приложений платформы. В случае каких-либо инцидентов, связанных с нарушением работы приложений, работающих в отказоустойчивом кластере, кластеру требуется некоторое время для восстановления работы, от нескольких секунд до нескольких десятков секунд. Во время восстановления Webiomed может работать с ошибками, но используемые механизмы отказоустойчивости позволяют автоматически и максимально быстро вернуть платформу к нормальной работе.

Одна из функций оркестратора - это обеспечение отказоустойчивости приложений за счет поддержания в работе необходимого количества экземпляров контейнеров и балансировки нагрузки между ними. Но отказоустойчивость средствами оркестратора возможна только для так называемых stateless-приложений (приложений без сохранения состояния), т.е. которые сами не хранят данные. К ним, например, относятся практически все приложения, обеспечивающие функционал системы. Есть еще так называемые statefull-приложения, которые как раз хранят данные, например, СУБД. Для обеспечения целостности данных для них не применим механизм масштабирования как для stateless-приложений. Отказоустойчивость для таких приложений, как правило, обеспечивается путем создания кластера встроенными средствами приложений. Для отказоустойчивости все системные statefull-приложения создаются в кластерах, состоящих из 3 узлов, что обеспечивает работоспособность кластера при выходе из строя одного узла. Для большей надежности все члены кластера должны быть установлены на разных серверах, чтобы при выходе из строя самого сервера отключалось минимальное количество узлов кластера приложения.

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата	
Изм.	Лист	№ докум.	Подпись	Дата	Лист
					9

Если необходимо сделать платформу отказоустойчивой, то для ее установки потребуется:

- Использовать установку только в режиме Docker Swarm;
- Выделить для установки системы не менее 3 серверов;
- Запланировать вычислительные ресурсы для масштабирования приложений, включая дисковое пространство, т.к. фактически выполняется трехкратное сохранение данных на разных узлах кластеров.

4.2.3 Требования к объему обрабатываемых данных

Работа Webiomed подразумевает постоянное накопление данных, которые хранятся в базах данных под управлением СУБД PostgreSQL. При малой нагрузке на систему можно ограничиться одним сервером СУБД под все базы платформы. Если нагрузка ожидается высокая, то самые нагруженные базы (dataset и dhra) следует разместить на отдельных серверах (кластерах СУБД, если требуется отказоустойчивость). Таким образом в системе будет не менее 3 серверов СУБД.

Распределение БД по разным серверам СУБД делается в первую очередь из соображений производительности. При этом сами экземпляры СУБД следует размещать на разных физических или виртуальных серверах. Также, пропорционально объему поступающих данных, увеличивается и количество операций по их обработке приложениями. Это в свою очередь потребует масштабирования в целях производительности ряда других приложений системы.

Соответственно, для создания системы, рассчитанной на большой объем обрабатываемых данных, потребуется:

- Использовать установку только в режиме Docker Swarm.
- Выделить для установки системы не менее 3 серверов.
- Запланировать вычислительные ресурсы для масштабирования приложений.

Име. № подл.	Подпись и дата
Взам. име. №	Име. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					10

4.2.4 Типы установок

На основании изложенных выше особенностей можно определить ряд типовых конфигураций установки Webiomed:

- **Тестовая.** Это минимальная установка на одном сервере с помощью Docker-compose, предназначенная для тестов или демонстрации работы Webiomed.
- **Базовая.** Это минимальная установка, рассчитанная на промышленное применение с низкой нагрузкой. Выполняется с помощью Docker Swarm на 1-2 серверах.
- **Отказоустойчивая.** Установка, в которой полностью обеспечена отказоустойчивость всех приложений. Выполняется с помощью Docker Swarm на 3 и более серверах. Т.к. в отказоустойчивой установке планируется не менее 2 экземпляров каждого приложения, то это обеспечивает еще и дополнительную производительность, т.е. установка будет рассчитана на достаточно высокий уровень нагрузки.
- **Отказоустойчивая для большого объема данных.** Установка, в которой важны как производительность, так и отказоустойчивость. Выполняется с помощью Docker Swarm на 3 и более серверах. При этом в системе будет 3 кластера СУБД и 9 узлов СУБД.

Необходимое количество серверов для каждого типа установки и их характеристики рассчитываются исходя из рекомендаций, описываемых в разделе 8.3 Масштабирование.

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата					
Изм.	Лист	№ докум.	Подпись	Дата					Лист
									11

4.3 Технические требования

4.3.1 Требования к оборудованию

Установка Webiomed может быть произведена на физические или виртуальные серверы. Независимо от использования виртуализации, физические серверы должны обладать следующими характеристиками:

Процессор архитектуры x86_64, частотой 2,0 ГГц и выше, с поддержкой инструкций AVX. Фактически для установки применимы практически все современные серверные процессоры. Инструкции AVX не поддерживают либо старые процессоры Intel (до 2013 года выпуска), либо бюджетные процессоры для настольных компьютеров и ноутбуков, например, линейки Intel Pentium.

Для промышленного применения дисковая подсистема должна обеспечивать производительность не ниже 3000 IOPS и основываться на отказоустойчивых массивах дисков. Такой уровень производительности обеспечивают дисковые массивы на твердотельных дисках.

Пропускная способность сети, в которой развернуты серверы для установки платформы, должна быть 1Gbit или выше.

Минимальные требования для установки в зависимости от типа:

Тестовая: 6 процессорных ядер, 24 ГБ оперативной памяти, 100 ГБ свободного дискового пространства.

Базовая: 8 процессорных ядер, 32 ГБ оперативной памяти, 500 ГБ свободного дискового пространства.

Отказоустойчивая: 20 процессорных ядер, 60 ГБ оперативной памяти, 1500 ГБ свободного дискового пространства.

Отказоустойчивая для большого объема данных: 50 процессорных ядер, 150 ГБ оперативной памяти, 4000 ГБ свободного дискового пространства.

Инв. № подл.	Подпись и дата				
	Инв. № дубл.				
	Взам. инв. №				
	Подпись и дата				
	Инв. № подл.				
Изм.	Лист	№ докум.	Подпись	Дата	Лист
					12

4.3.2 Требования к программному обеспечению

4.3.2.1 Требования к гипервизору

Для использования виртуальных серверов допускается применять любой гипервизор, поддерживающий рекомендуемые для использования операционные системы:

VMware vSphere/vSphere Hypervisor 6.7 или новее;

PVE 6.0 или новее.

4.3.2.2 Требования к операционным системам

Для установки системы требуется 64-разрядная ОС семейства Linux с ядром версии 3.10 или новее, командной оболочкой Bash версии 4.2 или новее, Python 3.6 или новее.

Работа системы проверена и поддерживается в следующих ОС:

- РЕД ОС «Сервер» 7.3 (<https://redos.red-soft.ru/product/server/>).
- Ubuntu Server 20.04 и 22.04 (<https://ubuntu.com/download/server>).

4.3.2.3 Требования к устанавливаемому ПО

Для работы системы требуется следующее ПО, устанавливаемое на серверах платформы:

Docker 19.03.0 или новее;

Docker-compose 1.25.5 или новее.

4.3.3 Другие требования

В системе не предусмотрена возможность прямой безопасной публикации серверов платформы в открытые сети. Все функции по публикации возлагаются на внешнее оборудование и/или ПО. К таким функциям относятся:

- Предоставление доступа по стандартным портам TCP для протоколов HTTP/HTTPS.
- Организация доступа по защищенному протоколу HTTPS и организация принудительного использования защищенного протокола HTTPS.
- Балансировка входящих запросов между серверами платформы.

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата					Лист
					Изм.	Лист	№ докум.	Подпись	Дата

Для публикации сервиса, например, в сеть Интернет, следует предусмотреть соответствующее оборудование и/или ПО. Это может быть связка из сетевого экрана и обратного прокси-сервера на базе Nginx (<http://nginx.org/>) или HAProxy (<http://www.haproxy.org/>). Либо может применяться специальное оборудование для организации защищенного доступа. Например, для организации доступа, защищаемого с помощью ГОСТ TLS, может применяться TLS-криптошлюз ViPNet TLS Gateway (<https://infotecs.ru/product/vipnet-tls-gateway.html>).

Инв. № подл.	Подпись и дата			Инв. № дубл.	Подпись и дата
	Взам. инв. №				
Изм.	Лист	№ докум.	Подпись	Дата	Лист 14

4.4 Планирование масштабирования

Под масштабированием понимается выделение и распределение для Webiomed необходимого объема вычислительных ресурсов (количество ядер процесса, объем ОЗУ, дискового пространства).

Обычно масштабирование делят на 2 категории: "вертикальное" — увеличение производительности отдельного вычислительного узла (сервера) платформы, например, путем увеличения объема ОЗУ, и "горизонтальное" — путем увеличения количества узлов платформы и распределения нагрузки между ними. Для построения надежной и производительной системы используются оба вида масштабирования.

В настоящее время уже редко используется установка серверного ПО прямо на физические серверы, как правило используется виртуализация, которая дает много преимуществ при масштабировании. При разворачивании платформы мы в первую очередь ориентируемся на то, что установка платформы для промышленного использования будет выполняться в виртуальных машинах и рекомендации по масштабированию даются исходя из особенностей использования виртуализации.

4.4.1 Особенности хранения данных

Данные приложений находятся в так называемых томах (volumes) Docker, которые используют только локальное дисковое устройство сервера. Это накладывает следующие ограничения:

- В целях повышения производительности, чтобы снизить конкуренцию за операции ввода-вывода, следует размещать сервисы, хранящие данных, либо на большем количестве серверов, либо подключать (монтировать) большее количество отдельных дисков к серверу.
- Приложения, работающие в отказоустойчивом кластере, следует размещать так, чтобы на одном диске сервера не находились данные двух и более членов кластера. Фактически это означает,

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата					Лист
					Изм.	Лист	№ докум.	Подпись	Дата

что для надежной работы требуется чтобы на одном сервере был только один член кластера.

Инв. № подл.	Подпись и дата			Инв. № дубл.	Подпись и дата
	Взам. инв. №				
Изм.	Лист	№ докум.	Подпись	Дата	Лист 16

5 Подготовка к установке

5.1 Установка необходимого ПО

5.1.1 Установка Docker

Установка Docker производится из официального репозитория с использованием последней стабильной версии.

5.1.1.1 Установка в РЕД ОС

1. Установите Docker и Docker-compose одной командой:

```
sudo dnf install docker-ce docker-compose
```

2. Включите автоматический запуск службы docker и запустите Docker:

```
sudo systemctl enable docker && sudo systemctl start docker
```

5.1.1.2 Установка в Ubuntu

1. Установите необходимые зависимости:

```
sudo apt-get update && \  
sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-  
release
```

2. Установите ключ репозитория:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \  
| sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

3. Настройте доступ к репозиторию:

```
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-  
keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs)  
stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

4. Установите docker:

```
sudo apt-get update && \  
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

- Все действия по установке одной командой:

```
sudo apt-get update && \  
sudo apt-get install -y apt-transport-https ca-certificates curl gnupg lsb-  
release && \  

```

Име. № подл.	Подпись и дата
Взам. инв. №	Име. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					17

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor
-o /usr/share/keyrings/docker-archive-keyring.gpg && \
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs)
stable" | sudo tee /etc/apt/sources.list.d/docker.list \> /dev/null && \
sudo apt-get update && \
sudo apt-get install -y docker-ce docker-ce-cli http://containerd.io
```

5.1.2 Установка Docker-compose

5.1.2.1 Установка в Ubuntu

1. Установите Python и необходимые для работы библиотеки:

```
sudo apt-get update && \
sudo apt-get install python3 python3-pip python3-wheel python3-setuptools
```

2. Обновите менеджер пакетов Pip:

```
sudo pip3 install -U pip
```

3. Установите Docker-compose:

```
sudo pip3 install docker-compose
```

Все действия по установке одной командой:

```
sudo apt-get update && \
sudo apt-get install -y python3 python3-pip \
python3-wheel python3-setuptools && \
sudo pip3 install -U pip && \
sudo pip3 install docker-compose
```

5.1.3 Настройка серверов

5.1.3.1 Настройки ядра

При установке платформы в Docker Swarm следует выполнить настройки операционной системы для повышения стабильности и производительности. Для этого на каждом сервере, предназначенном для работы платформы, в файле */etc/sysctl.conf* должны быть указаны следующие параметры:

```
net.ipv4.ip_local_port_range=20000 65000
net.ipv4.tcp_tw_reuse=1
net.ipv4.tcp_fin_timeout=15
net.core.somaxconn=4096
net.core.netdev_max_backlog=4096
net.core.rmem_max=16777216
```

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата					Лист
									18
					Изм.	Лист	№ докум.	Подпись	Дата

```
net.core.wmem_max=16777216
net.ipv4.tcp_max_syn_backlog=20480
net.ipv4.tcp_max_tw_buckets=400000
net.ipv4.tcp_no_metrics_save=1
net.ipv4.tcp_rmem=4096 87380 16777216
net.ipv4.tcp_syn_retries=2
net.ipv4.tcp_synack_retries=2
net.ipv4.tcp_wmem=4096 65536 16777216
net.netfilter.nf_conntrack_max=262144
net.netfilter.nf_conntrack_tcp_timeout_established=86400
```

Для применения настроек выполните команду:

```
sudo sysctl -p
```

5.2 Настройка кластера Docker Swarm

5.2.1 Общие сведения

Перед настройкой кластера следует определить роли узлов в кластере. На master-узлах происходит управление кластером: выполняются команды конфигурирования кластером и создаются задания для разворачивания сервисов Docker на worker-узлах.

Worker-узлы предназначены только запуска сервисов Docker. По умолчанию master-узлы также могут работать как worker-узлы, т.е. могут быть задействованы для установки приложений.

Master-узлов может быть несколько для отказоустойчивости. В этом случае master-узлы автоматически определяют ведущий узел, используя протокол Raft (<https://raft.github.io/>) и синхронизируют данные о состоянии кластера. Для корректной работы распределенных master-узлов требуется, чтобы их было нечетное количество, обычно 3 или 5. При 3 узлах управление кластером сохраняется при выходе из строя одного любого узла, при 5 узлах допускается потеря любых 2.

При использовании отказоустойчивого управления кластером, при 3 или 5 master-узлах, не следует использовать их для размещения приложения, для исключения их влияния на работу протокола Raft.

Для создания и работы кластера Docker Swarm необходим как минимум один сервер. Кластер можно как расширять, добавляя master- и worker-узлы,

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					19

так и сокращать в процесс эксплуатации без прекращения его работы и запущенных приложений.

Для назначения узлов, на которых могут работать те или иные приложения платформы, используются так называемые метки узлов. Полный список используемых меток приведен в п. 10.3. На узлах без меток приложения платформы не будут устанавливаться. При установке меток следует следить за уникальностью меток для statefull-приложений, чтобы не было узлов с одинаковыми метками, т.к. это может привести к потере данных и нарушению работы платформы.

При любом изменении состава кластера следует учитывать, что данные приложений не могут автоматически перемещаться между узлами. При смене узла работающего statefull-приложения придется самостоятельно переносить его данные.

5.2.2 Создание master-узла

На сервере, который будет первым master-узлом, выполните команду следующего вида, которая инициализирует кластер:

```
docker swarm init --advertise-addr <IP-адрес сервера>
```

Например:

```
sudo docker swarm init --advertise-addr 192.168.99.121
```

В результате работы команды будет выведена команда для подключения worker-узлов в кластер, содержащая токен присоединения к кластеру. Она будет в дальнейшем использоваться для подключения worker-узлов, поэтому рекомендуется ее сохранить.

5.2.3 Подключение worker-узлов к кластеру

На всех подключаемых worker-узлах выполните команду для подключения к кластеру, полученную при создании master-узла. Если она не известна, то получить ее заново можно, выполнив на master-узле следующую команду:

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата					Лист
					Изм.	Лист	№ докум.	Подпись	Дата

```
sudo docker swarm join-token worker
```

После подключения узла проверьте состояние кластера, выполнив на master-узле команду:

```
sudo docker node ls
```

При успешном подключении узлов к кластеру они все должны присутствовать в списке и быть в состоянии «Active» (рис.1).

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS
krdbdwe2jqrs6vuu5u8s7nwev *	CONTROL-1	Ready	Active	Leader
6gba5w39t8lwn615s4ykrhxy	DATASTORE-1	Ready	Active	
ls3l54la4fpo55nt9cil0oygg	WEBIOMED-APP-1	Ready	Active	
nl0cw9dzco7qpol6ds3qvgr8a	WEBIOMED-APP-2	Ready	Active	
yjk8ruuwfva3wec1j9zatnws1	WEBIOMED-APP-3	Ready	Active	
nv2t92vyiwnp68ulnvk0veoo	WEBIOMED-DB-1	Ready	Active	
rlusskg799clqt9evdgc5pzz	WEBIOMED-DB-2	Ready	Active	
mbxa9j97f1tuqumg6lpx5q5y	WEBIOMED-DB-3	Ready	Active	

Рисунок 1. Проверка состояния кластера.

5.3 Подготовка дистрибутива платформы

1. Получите архив дистрибутива и загрузите его на master-сервер (для установки в Docker Swarm) или на сервер системы в иных случаях.
2. Распакуйте архив, выполнив из каталога с файлом дистрибутива команду следующего вида:

```
sudo unzip -d /opt <имя файла>
```

Например:

```
sudo unzip -d /opt webiomed.zip
```

После распаковки файлы дистрибутива будут находиться в каталоге `/opt/webiomed`.

Име. № подл.	Подпись и дата
Взам. инв. №	Име. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					21

6 Установка

6.1 Общее описание

Установка Webiomed в общем случае состоит из 3 этапов:

1. Установка локального реестра образов. Требуется, если установка Webiomed выполняется в Docker Swarm, в остальных случаях не обязательна.
2. Установка Webiomed.
3. Установка инструментов мониторинга.

Установка должна выполняться именно в указанном порядке: вначале реестр образов, затем Webiomed, последними инструменты мониторинга.

Все операции по установке выполняются на master-сервере при установке в Docker Swarm или на сервере системы в иных случаях.

6.2 Установка локального реестра образов

6.2.1 Установка

1. Перейдите в каталог с распакованным дистрибутивом системы:

```
cd /opt/webiomed
```

2. Выполните команду, которая подготовит конфигурационные файлы системы и выполнит проверку готовности к установке:

```
sudo ./webiomed.sh preinstall
```

3. Выполните первичную настройку конфигурационных файлов. Для установки Docker Registry как минимум необходимо в файле `/etc/webiomed/system.conf` установить значение параметра `SWARM_ENABLED`. При значении `true` установка будет проходить в режиме Docker Swarm, при `false` в Docker Compose.

Если выбран режим Docker Swarm (`SWARM_ENABLED=true`), то после сохранения файла `system.conf` следует назначить сервер на котором будут размещаться данные реестра образов. Для этого следует установить метку узла кластера Docker Swarm, выполнив команду вида:

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата					Лист
					Изм.	Лист	№ докум.	Подпись	Дата

```
sudo docker node update --label-add registry=true <имя узла swarm>
```

Например:

```
sudo docker node update --label-add registry=true server-1
```

Список доступных узлов можно получить с помощью команды:

```
sudo docker node ls
```

4. Снова выполните команду для проверки готовности к установке:

```
sudo ./webiomed.sh preinstall
```

5. Если ошибок и предупреждений нет, то запустите установку

Registry:

```
sudo ./registry.sh install
```

6.2.2 Настройки для использования Registry

1. В файле `/etc/webiomed/system.conf` раскомментируйте значение параметра `REGISTRY_HOST`, чтобы он выглядел следующим образом:

```
REGISTRY_HOST=localhost:5000/
```

2. На каждом узле кластера Docker Swarm необходимо в файле `/etc/docker/daemon.json` включить адрес реестра образов в список доверенных узлов. Для этого необходимо в файл `/etc/docker/daemon.json` добавить следующее содержимое:

```
{  
  "insecure-registries" : ["localhost:5000"]  
}
```

3. После сохранения файла перезапустите Docker с помощью команды:

```
sudo service docker restart
```

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата					Лист
					Изм.	Лист	№ докум.	Подпись	Дата

6.3 Установка платформы

1. Перейдите в каталог с распакованным дистрибутивом Webiomed:

```
cd /opt/webiomed
```

2. Выполните команду, которая подготовит конфигурационные файлы системы и выполнит проверку готовности к установке:

```
sudo ./webiomed.sh preinstall
```

3. Выполните первичную настройку конфигурационных файлов. Для установки платформы необходимо в файле */etc/webiomed/system.conf* установить значение следующих параметров:

- SWARM_ENABLED. При значении true установка будет проходить в режиме Docker Swarm, при false в Docker Compose.
- REGISTRY_HOST. Обязательно следует установить значение (раскомментировать) при установке в Docker Swarm. Также предварительно должен быть установлен сам Docker Registry.
- HA_ENABLED. При значении true установка приложений Redis и RabbitMQ будет произведена в режиме кластера. По умолчанию false, устанавливается по одному экземпляру приложений.
- POSTGRES_HA. При значении true установка СУБД PostgreSQL будет произведена в режиме кластера. По умолчанию false, устанавливается один сервер СУБД.
- BIGDATA_ENABLED. При значении true для баз данных сервисов DHRA и Dataset будут созданы отдельные экземпляры СУБД, а брокер сообщений Apache Kafka будет установлен в режиме кластера. По умолчанию false, все базы данных устанавливаются на один сервер СУБД.
- UNSECURE_COOKIES_POLICY. Определяет возможность использования небезопасных кук. По умолчанию false (небезопасные куки отключены) - система рассчитана на

Инв. № подл.	Подпись и дата				Лист 24
	Инв. № дубл.				
	Взам. инв. №				
	Подпись и дата				
	Подпись и дата				
Изм.	Лист	№ докум.	Подпись	Дата	

использование безопасного протокола HTTPS. Если планируется использовать доступ по небезопасному протоколу HTTP, например, для настройки внутри доверенной локальной сети, следует установить значение true, включив небезопасные куки.

4. В каталоге /etc/webiomed/secrets находятся файлы настроек приложений, содержащие различные пароли и ключи безопасности. Если их не установить самостоятельно перед установкой, то они будут сгенерированы случайным образом при установке. При необходимости, можно установить необходимые значения ключей и паролей, описание параметров конфигурационных файлов приведено в п. 10.4.
5. Если выбран режим установки в Docker Swarm, то следует назначить серверы для размещения приложений путем установки меток узлов кластера Docker Swarm. Для этого следует выполнить команду следующего вида:

```
sudo docker node update --label-add <метка> <имя узла swarm>
```

Список доступных узлов можно получить с помощью команды:

```
sudo docker node ls
```

Полный список назначаемых меток приведён в п. 10.3.

6. Снова выполните команду для проверки готовности к установке:

```
sudo ./webiomed.sh preinstall
```

7. Если ошибок и предупреждений нет, то запустите установку Webiomed:

```
sudo ./webiomed.sh install
```

Ине. № подл.	Подпись и дата
Взам. инв. №	Ине. № дубл.
Подпись и дата	Ине. № подл.

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					25

6.4 Установка инструментов мониторинга

1. Если платформа устанавливается в Docker Swarm, то следует установить метку на узел кластера, на котором будут размещаться приложения мониторинга. Для этого следует выполнить команду следующего вида:

```
sudo docker node update monitoring=true <имя узла swarm>
```

Список доступных узлов можно получить с помощью команды:

```
sudo docker node ls
```

2. Перейдите в каталог с распакованным дистрибутивом системы:

```
cd /opt/webiomed
```

3. Запустите установку инструментов мониторинга:

```
sudo ./monitoring.sh install
```

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата					Лист
									26
					Изм.	Лист	№ докум.	Подпись	Дата

7 Настройка

7.1 Настройка системных параметров платформы

Настройка системных параметров платформы производится путем установки необходимых значений параметров в файле `/etc/webiomed/system.conf`, в котором указываются общие для платформы настройки, и в файлах в каталоге `/etc/webiomed/secrets`, которых указываются индивидуальные настройки приложений. Полное описание возможных параметров приведено в п. 10.4.

Для изменения настроек требуется установить необходимые значения в файлах. Для применения настроек выполните следующую команду из каталога с распакованным дистрибутивом платформы:

```
sudo ./webiomed.sh start
```

7.2 Настройка резервного копирования

В платформе предусмотрено выполнение автоматического резервного копирования баз данных платформы, выполняемого с использованием утилиты `pg_dump` СУБД PostgreSQL. Для настройки резервного копирования откройте в текстовом редакторе файл системных настроек `/etc/webiomed/system.conf` и установите значения следующих параметров:

- `PGBACKUP_ENABLED`. Установите значение `true` для включения резервного копирования по расписанию. По умолчанию `false` (резервное копирование не выполняется).
- `PGBACKUP_DIR`. Каталог для хранения резервных копий, по умолчанию `/backup`.
- `PGBACKUP_SCHEDULE`. Расписание запуска резервного копирования в формате `crontab` в часовой зоне UTC. По умолчанию `"0 23 * * *"` (ежедневно в 23:00 UTC).
- `PGBACKUP_FORMAT`. Формат резервных копий, допустимо использовать `custom` или `directory`, по умолчанию используется формат `custom`.

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата					Лист	
									27	
					Изм.	Лист	№ докум.	Подпись	Дата	

- PGBACKUP_JOBS. Количество потоков копирования, по умолчанию 1. Увеличение количества потоков сокращает время резервного копирования, но требует дополнительных процессорных ресурсов из расчета 1 ядро на поток. При значении 2 и более требуется установить параметр PGBACKUP_FORMAT=directory.
- PGBACKUP_KEEP_FILES. Определяет интервал ротации резервных копий (сколько копий хранить), по умолчанию хранится 3 последних копии.
- PGBACKUP_COMPRESS. Определяет уровень сжатия файлов резервных копий. Возможны значения от 1 (минимальное сжатие) до 9 (максимальное сжатие), по умолчанию 3. Чем выше значение, тем меньше размер файла, но тем дольше тратится времени на резервное копирование.

Пример настроек резервного копирования:

```
PGBACKUP_ENABLED=true
PGBACKUP_DIR=/pgbackup
PGBACKUP_SCHEDULE=0 17 * * *
PGBACKUP_FORMAT=directory
PGBACKUP_JOBS=2
PGBACKUP_KEEP_FILES=3
PGBACKUP_COMPRESS=4
```

Для применения настроек перейдите в каталог с распакованным дистрибутивом и выполните команду:

```
sudo ./webiomed.sh start
```

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата					Лист
									28
					Изм.	Лист	№ докум.	Подпись	Дата

8 Обслуживание

8.1 Управление сервисами платформы

Все управление сервисами платформы осуществляется с помощью установочных скриптов, поставляемых с дистрибутивом системы:

- `webiomed.sh` – для управления платформой;
- `monitoring.sh` – для управления инструментами мониторинга;
- `registry.sh` – для управления локальном реестром образов.

Для выполнения каких-либо действий необходимо на master-сервере (в режиме Docker Swarm) или на самом сервере, где установлен Webiomed, выполнить следующие действия:

1. Перейдите в каталог с распакованным архивом дистрибутива:

```
cd /opt/webiomed
```

2. Выполните команду управления следующего вида:

```
sudo ./<файл скрипта> <команда> <параметры команды>
```

Полный список параметров скриптов и их описание приведено в п. 10.2. Например, для перезагрузки только сервиса API приложения Dataset необходимо выполнить команду:

```
sudo ./webiomed.sh reload dataset
```

8.2 Обновление

1. Получите архив дистрибутива новой версии платформы и загрузите его на master-сервер (для установки в Docker Swarm) или на сервер системы в иных случаях.
2. Удалите файлы предыдущей установки:

```
sudo rm -rf /opt/webiomed
```

3. Распакуйте архив, выполнив из каталога с файлом дистрибутива команду следующего вида:

```
sudo unzip -d /opt <имя файла>
```

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата					Лист
									29
Изм.	Лист	№ докум.	Подпись	Дата					

Например:

```
sudo unzip -d /opt webiomed.zip
```

После распаковки файлы дистрибутива будут находиться в каталоге `/opt/webiomed`.

4. Перейдите в каталог с распакованным дистрибутивом системы:

```
cd /opt/webiomed
```

5. Запустите процедуру обновления платформы командой:

```
sudo ./webiomed.sh upgrade
```

6. Запустите процедуру обновления инструментов мониторинга командой:

```
sudo ./monitoring.sh upgrade
```

8.3 Масштабирование

При установке в Docker Swarm возможно масштабирование сервисов платформы в целях повышения производительности и отказоустойчивости за счет увеличения количества одновременно работающих контейнеров. По умолчанию устанавливается только один контейнер на каждый сервис.

Изменение количества контейнеров сервисов платформы выполняется путем установки необходимого числа в параметрах файла настроек `/etc/webiomed/system.conf`. Формат параметра, задающего количество контейнеров для какого-либо сервиса, имеет следующий вид:

```
SERVICE_<имя сервиса>_REPLICAS=<количество>
```

Полный список сервисов платформы, включая их имена, приведен в п. 10.1. При составлении имени параметра следует сделать следующие преобразования:

- 1) Перевести все символы в верхний регистр.
- 2) Заменить символ «-» на «_».

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					30

Например, требуется увеличить количество контейнеров сервиса API приложения Symptom-checker, у которого имя в системе symptom-checker. В этом случае, если требуется 4 контейнера, следует добавить в файл /etc/webiomed/system.conf параметр:

```
SERVICE_SYMPTOM_CHECKER_REPLICAS=4
```

Для применения настроек перейдите в каталог с распакованным дистрибутивом платформы и выполните команду:

```
sudo ./webiomed.sh start
```

Проконтролировать состояние масштабирования можно с помощью команды:

```
sudo docker service ls
```

В колонке «REPLICAS» будет показано текущее состояние развертывания сервиса в формате <Количество запущенных контейнеров>/<Необходимое количество контейнеров> (рис.2).

ID	NAME	MODE	REPLICAS
oim65ziv5g18	monitoring_alertmanager	replicated	1/1
wg4ayww4md3l	monitoring_cadvisor	global	8/8
9v30ikaih99b	monitoring_grafana	replicated	1/1
lqbyecij4gmc	monitoring_karma	replicated	1/1
xyfvj41cv73t	monitoring_loki	replicated	1/1
wg8xbv90brsk	monitoring_node-exporter	global	8/8
tp5jl66498ym	monitoring_portainer	replicated	1/1
qyut606w34za	monitoring_portainer-agent	global	8/8
tbh7yg9e5h6q	monitoring_prometheus	replicated	1/1
pyhj7g62u8ng	monitoring_promtail	global	8/8
gqpquutt12mar	monitoring_pushgateway	replicated	1/1
v0jezwneqbj	monitoring_victoria-metrics	replicated	1/1
dszld19vlf6u	registry_registry	replicated	1/1
s9dbbetotlju	webiomed_blackbox-exporter	replicated	1/1
ettday6gafh8	webiomed_dataset	replicated	8/8
a3beapii9pd7	webiomed_dataset-consumer-calculated-signs	replicated	1/1
jfpv0tmap8us	webiomed_dataset-consumer-dispensary-observation-sending	replicated	3/3
x47qy60nxeaq	webiomed_dataset-consumer-extracting-patients-diagnoses	replicated	1/1
yaugff5iihlx	webiomed_dataset-consumer-laboratory-research	replicated	1/1
yd70vx0a2rec	webiomed_dataset-consumer-nlp-mass-documents-result	replicated	3/3
u547o1aefmpu	webiomed_dataset-consumer-patients-detail-results	replicated	4/4
a71pv5pjk0em	webiomed_dataset-consumer-patients-diseases	replicated	2/2
ms5vxn8f6epx	webiomed_dataset-consumer-re-extract-nlp-mass-documents-result	replicated	3/3
62tm3g7xr2sq	webiomed_dataset-consumer-request-ai-sending	replicated	11/11

Рисунок 2. Состояние масштабирования.

Име. № подл.	Подпись и дата	Взам. име. №	Име. № дубл.	Подпись и дата					Лист
									31
					Изм.	Лист	№ докум.	Подпись	Дата

8.4 Резервное копирование

8.4.1 Ручное создание резервных копий

Для ручного создания резервной копии базы данных платформы необходимо выполнить на сервере, назначенным для резервного копирования, команду следующего вида:

```
sudo docker exec \  
$(sudo docker ps --filter "name=webiomed_pgbackup" --format "{{.ID}}") \  
/scripts/pgbackup.sh <список БД>
```

Если не указать имена баз данных, то выполнится резервное копирование всех баз. Например, для резервного копирования, только баз данных приложений DHRA и Dataset выполните команду:

```
sudo docker exec \  
$(sudo docker ps --filter "name=webiomed_pgbackup" --format "{{.ID}}") \  
/scripts/pgbackup.sh dhra dataset
```

8.4.2 Восстановление

Для восстановления базы данных платформы из резервной копии необходимо выполнить на сервере, назначенным для резервного копирования, команду следующего вида:

```
sudo docker exec \  
$(sudo docker ps --filter "name=webiomed_pgbackup" --format "{{.ID}}") \  
/scripts/pgrestore.sh <файл резервной копии>
```

По умолчанию резервные копии хранятся в каталоге, назначаемом в параметре `PGBACKUP_DIR` в файле системных настроек `/etc/webiomed/system.conf`, по умолчанию в каталог `/backup`. Список доступных резервных копий можно получить с помощью команды:

```
sudo ls -l /backup
```

Имена файлов резервных копий имеют следующий формат:

<имя БД>_<дата создания копии>_<время создания копии>.bak

Име. № подл.	Подпись и дата
Взам. име. №	Име. № дубл.
Подпись и дата	

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					32

Например, для восстановления баз данных приложения Gate, используя файл резервной копии *gate_2023-01-22_23-00-01UTC.bak*, выполните следующую команду:

```
sudo docker exec \
$(sudo docker ps --filter "name=webiomed_pgbackup" --format "{{.ID}}") \
/scripts/pgrestore.sh gate_2023-01-22_23-00-01UTC.bak
```

Инв. № подл.	Подпись и дата			Подпись и дата		Лист
	Взам. инв. №			Инв. № дубл.		
	Подпись и дата			Инв. № дубл.		
	Подпись и дата			Инв. № дубл.		
Изм.	Лист	№ докум.	Подпись	Дата	33	

9 Удаление платформы

Для удаления платформы в первую очередь требуется остановить все сервисы платформы. Для это из каталога с распакованным дистрибутивом платформы выполните следующие команды:

```
sudo ./monitoring.sh stop
sudo ./webiomed.sh stop
sudo ./registry.sh stop
```

Для удаления оставшихся данных платформы на каждом сервере, используемом для работы платформы, выполните команду:

```
sudo docker system prune --volumes --force
```

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата
Изм.	Лист	№ докум.	Подпись	Дата

<i>Лист</i>
34

10 Приложения

10.1 Сервисы платформы

Таблица 1. Сервисы локального реестра образов.

№	Модуль/Группа	Название сервиса	Назначение
1	Инфраструктурные сервисы	registry	Приложение реестра образов

Таблица 2. Сервисы Webiomed.

№	Модуль/Группа	Название сервиса	Назначение
1	Dataset	dataset	API приложения Dataset
2	Dataset	dataset-consumer-calculated-signs	Консьюмер Apache Kafka приложения Dataset
3	Dataset	dataset-consumer-dispensary-observation-sending	Консьюмер Apache Kafka приложения Dataset
4	Dataset	dataset-consumer-extracting-patients-diagnoses	Консьюмер Apache Kafka приложения Dataset
5	Dataset	dataset-consumer-laboratory-research	Консьюмер Apache Kafka приложения Dataset
6	Dataset	dataset-consumer-nlp-mass-documents-result	Консьюмер Apache Kafka приложения Dataset
7	Dataset	dataset-consumer-patients-detail-results	Консьюмер Apache Kafka приложения Dataset
8	Dataset	dataset-consumer-patients-diseases	Консьюмер Apache Kafka приложения Dataset
9	Dataset	dataset-consumer-re-extract-nlp-mass-documents-result	Консьюмер Apache Kafka приложения Dataset
10	Dataset	dataset-consumer-request-ai-sending	Консьюмер Apache Kafka приложения Dataset
11	Dataset	dataset-tasks-monitoring	Приложение мониторинга выполнения регулярных задач Dataset
12	Dataset	dataset-tasks-scheduler	Планировщик задач Dataset

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					35

13	Dataset	dataset-tasks-workers	Исполнители задач Dataset
14	DHRA	dhra	Консьюмер Apache Kafka приложения DHRA
15	DHRA	dhra-consumer-patients-reassessment-result	Консьюмер Apache Kafka приложения DHRA
16	DHRA	dhra-consumer-requests-group-updating	Консьюмер Apache Kafka приложения DHRA
17	DHRA	dhra-tasks-monitoring	Приложение мониторинга выполнения регулярных задач DHRA
18	DHRA	dhra-tasks-scheduler	Планировщик задач DHRA
19	DHRA	dhra-tasks-workers	Исполнители задач DHRA
20	DMA	dma	API приложения DMA
21	DMA	dma-cron	Приложение для запуска задач DMA по расписанию
22	Gate	gate	API приложения Gate
23	NLP	nlp	API приложения NLP
24	NLP	nlp-consumer-dataset-nlp-mass-documents	Консьюмер Apache Kafka приложения NLP
25	NLP	nlp-consumer-dataset-re-extract-nlp-mass-documents	Консьюмер Apache Kafka приложения NLP
26	PatientID	patientid	API приложения PatientID
27	PatientID	patientid-consumer-patients-calculating-signs	Консьюмер Apache Kafka приложения PatientID
28	PatientID	patientid-consumer-patients-detail	Консьюмер Apache Kafka приложения PatientID
29	PatientID	patientid-consumer-patients-group-updating	Консьюмер Apache Kafka приложения PatientID
30	Symptom-checker	symptom-checker	API приложения Symptom-checker
31	Symptom-checker	symptom-checker-consumer-patients_reassessment	Консьюмер Apache Kafka приложения Symptom-checker

Ине. № подл.	Подпись и дата
Взам. инв. №	Ине. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					36

32	Symptom-checker	symptom-checker-directory	API справочников приложения Symptom-checker
33	Веб-интерфейс	endpoint	Веб-интерфейс системы
34	Веб-интерфейс	frontend	Веб-интерфейс системы
35	Веб-интерфейс	frontend-dhra	Веб-интерфейс системы версии 1.0
36	Системные сервисы	ingress	Маршрутизатор http-запросов в систему
37	Системные сервисы	postgres-1	СУБД PostgreSQL
38	Системные сервисы	postgres-2	СУБД PostgreSQL
39	Системные сервисы	postgres-3	СУБД PostgreSQL
40	Системные сервисы	postgres-dataset-1	СУБД PostgreSQL
41	Системные сервисы	postgres-dataset-2	СУБД PostgreSQL
42	Системные сервисы	postgres-dataset-3	СУБД PostgreSQL
43	Системные сервисы	postgres-dhra-1	СУБД PostgreSQL
44	Системные сервисы	postgres-dhra-2	СУБД PostgreSQL
45	Системные сервисы	postgres-dhra-3	СУБД PostgreSQL
46	Системные сервисы	postgres-exporter	Экспортер метрик СУБД PostgreSQL
47	Системные сервисы	postgres-db-exporter	Экспортер метрик баз данных СУБД PostgreSQL
48	Системные сервисы	rabbitmq-1	Брокер сообщений RabbitMQ
49	Системные сервисы	rabbitmq-2	Брокер сообщений RabbitMQ
50	Системные сервисы	rabbitmq-3	Брокер сообщений RabbitMQ
51	Системные сервисы	redis-1	СУБД Redis
52	Системные сервисы	redis-2	СУБД Redis
53	Системные сервисы	redis-3	СУБД Redis

Име. № подл.	Взам. инв. №	Инв. № дубл.	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					37

54	Системные сервисы	redis-sentinel-1	Приложение Redis-sentinel для контроля за репликами СУБД Redis
55	Системные сервисы	redis-sentinel-2	Приложение Redis-sentinel для контроля за репликами СУБД Redis
56	Системные сервисы	redis-sentinel-3	Приложение Redis-sentinel для контроля за репликами СУБД Redis
57	Системные сервисы	redis-exporter	Экспортер метрик СУБД Redis
58	Системные сервисы	zookeeper-1	Распределенная система хранения конфигураций Zookeeper
59	Системные сервисы	zookeeper-2	Распределенная система хранения конфигураций Zookeeper
60	Системные сервисы	zookeeper-3	Распределенная система хранения конфигураций Zookeeper
61	Системные сервисы	kafka-1	Распределенный брокер сообщений Apache Kafka.
62	Системные сервисы	kafka-2	Распределенный брокер сообщений Apache Kafka.
63	Системные сервисы	kafka-3	Распределенный брокер сообщений Apache Kafka.
64	Системные сервисы	haproxy	Балансировщик нагрузки HAProxy
65	Системные сервисы	docker-gc	Приложение очистки серверов от неиспользуемых контейнеров и образов
66	Системные сервисы	kafka-manager	Инструмент администрирования Apache Kafka
67	Системные сервисы	pgadmin4	Инструмент администрирования СУБД PostgreSQL
68	Системные сервисы	pgbackup	Приложение для резервного копирования баз данных СУБД PostgreSQL

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					38

Таблица 3. Сервисы мониторинга.

№	Модуль/Группа	Название сервиса	Назначение
	Мониторинг	alertmanager	Alertmanager (https://github.com/prometheus/alertmanager) - отвечает за рассылку уведомлений о событиях мониторинга
	Мониторинг	cadvisor	Cadvisor (https://github.com/google/cadvisor) - экспортер метрик работы контейнеров приложений
	Мониторинг	grafana	Grafana (https://grafana.com/oss/grafana) - средство визуализации данных мониторинга
	Мониторинг	karma	Karma (https://github.com/primitive/karma) - дашборд для Alertmanager, отображает текущие события мониторинга, требующие внимания
	Мониторинг	loki	Loki (https://github.com/grafana/loki) - система агрегации логов
	Мониторинг	node-exporter	Node exporter (https://github.com/prometheus/node_exporter) - экспортер метрик операционной системы и оборудования сервера
	Мониторинг	portainer	Portainer (https://github.com/portainer/portainer) - серверная часть веб-интерфейса для управления контейнерами приложений
	Мониторинг	portainer-agent	Portainer Agent (https://github.com/portainer/agent) - агент Portainer
	Мониторинг	prometheus	Prometheus (https://github.com/prometheus/prometheus) - система мониторинга
	Мониторинг	promtail	Promtail (https://github.com/grafana/loki) -

Име. № подл.	Подпись и дата
Взам. име. №	Име. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

			агент сбора логов и отправки их в Loki
	Мониторинг	pushgateway	Pushgateway (https://github.com/prometheus/pushgateway) - шлюз для самостоятельной отправки метрик приложениями в Prometheus
	Мониторинг	victoria-metrics	VictoriaMetrics (https://github.com/VictoriaMetrics/VictoriaMetrics) - СУБД для хранения данных метрик

10.2 Параметры установочных скриптов

Таблица 4. Параметры установочного скрипта локального реестра образов registry.sh

№	Параметр	Описание
1	login	Подключение к реестру образов (ввод учетных данных в реестре образов)
2	install	Новая установка реестра образов
3	upgrade <список сервисов>	Обновление указанных сервисов. Если сервисы не указаны, то выполняется обновление всех приложений.
4	delete	Удалить платформу
5	stop	Остановить все сервисы
6	start	Запустить все сервисы
7	update	Обновить состояние всех сервисов. Эквивалент параметра start
8	restart	Перезапустить все сервисы
9	reload <список сервисов>	Перезапустить указанные сервисы. Если сервисы не указаны, то перезапускаются все сервисы.
10	services	Вывести список доступных сервисов
11	help	Вывод справки по параметрам

Име. № подл.	Подпись и дата
Взам. име. №	Подпись и дата
Име. № дубл.	Подпись и дата
Име. № дубл.	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					40

Таблица 5. Параметры установочного скрипта платформы webiomed.sh

№	Параметр	Описание
1	login	Подключение к реестру образов (ввод учетных данных в реестре образов)
2	preinstall	Подготовка к установке и проверка готовности к установке
3	build <список сервисов>	Загрузка образов и сохранение образов в локальном реестре образов
4	install	Новая установка платформы
5	upgrade <список сервисов>	Обновление указанных сервисов. Если сервисы не указаны, то выполняется обновление всех приложений.
6	post-upgrade	Выполнить только предусмотренные задачи после обновления.
7	delete	Удалить платформу
8	stop	Остановить все сервисы
9	start	Запустить все сервисы
10	update	Обновить состояние всех сервисов. Эквивалент параметра start
11	restart	Перезапустить все сервисы
12	reload <список сервисов>	Перезапустить указанные сервисы. Если сервисы не указаны, то перезапускаются все сервисы.
13	services	Вывести список доступных сервисов
14	update-system	Запуск/обновление состояния только системных сервисов
15	help	Вывод справки по параметрам

Таблица 6. Параметры установочного скрипта инструментов мониторинга monitoring.sh.

№	Параметр	Описание
1	login	Подключение к реестру образов (ввод учетных данных в реестре образов)
2	build <список сервисов>	Загрузка образов и сохранение образов в локальном реестре образов
3	install	Новая установка инструментов мониторинга
4	upgrade <список сервисов>	Обновление указанных сервисов. Если сервисы не указаны, то выполняется обновление всех приложений.
5	delete	Удалить платформу
6	stop	Остановить все сервисы
7	start	Запустить все сервисы

Име. № дубл.	Подпись и дата
Взам. инв. №	Подпись и дата
Име. № подл.	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

8	update	Обновить состояние всех сервисов. Эквивалент параметра start
9	restart	Перезапустить все сервисы
10	reload <список сервисов>	Перезапустить указанные сервисы. Если сервисы не указаны, то перезапускаются все сервисы.
11	services	Вывести список доступных сервисов
12	help	Вывод справки по параметрам

10.3 Метки узлов кластера Docker Swarm

Таблица 7. Метки локального реестра образов.

№	Название метки	Уникальная (только на одном узле)	Назначение
1	registry=true	да	Узел для размещения реестра образов

Таблица 8. Метки приложений платформы.

№	Название метки	Уникальная (только на одном узле)	Назначение
1	webiomed_app=true	нет	Узлы для размещения приложений платформы
2	webiomed_dma=true	да	Узел для размещения приложения DMA
3	webiomed_kafka_1=true	да	Узел для размещения Apache Kafka. При установке в кластере узел №1 кластера.
4	webiomed_kafka_2=true	да	Узел для размещения узла №2 кластера Apache Kafka
5	webiomed_kafka_3=true	да	Узел для размещения узла №3 кластера Apache Kafka
6	webiomed_pgadmin4=true	да	Узел для размещения приложения PgAdmin4
7	webiomed_pgbackup=true	да	Узел для размещения приложения резервного копирования СУБД
8	webiomed_postgres_1=true	да	Узел для размещения сервера СУБД

Име. № подл.	Подпись и дата
	Име. № дубл.
Изм. № док.	Взам. име. №
	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					42

			PostgreSQL. При установке в кластере узел №1 кластера
9	webiomed_postgres_2=true	да	Узел для размещения узла № 2 СУБД PostgreSQL при установке в кластере
10	webiomed_postgres_3=true	да	Узел для размещения узла № 3 СУБД PostgreSQL при установке в кластере
11	webiomed_postgres_dataset_1=true	да	Узел для размещения узла СУБД PostgreSQL для БД приложения Dataset при установке для большого объема данных. При установке в кластере узел №1 кластера для БД приложения Dataset.
12	webiomed_postgres_dataset_2=true	да	Узел для размещения узла № 2 СУБД PostgreSQL для БД приложения Dataset при установке в кластере.
13	webiomed_postgres_dataset_3=true	да	Узел для размещения узла № 3 СУБД PostgreSQL для БД приложения Dataset при установке в кластере.
14	webiomed_postgres_dhra_1=true	да	Узел для размещения узла СУБД PostgreSQL для БД приложения DHRA при установке для большого объема данных. При установке в кластере узел №1 кластера для БД приложения DHRA.
15	webiomed_postgres_dhra_2=true	да	Узел для размещения узла № 2 СУБД PostgreSQL для БД приложения DHRA при установке в кластере.

Ине. № подл.	Подпись и дата
Взам. инв. №	Ине. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

16	webiomed_postgres_dhra_3=true	да	Узел для размещения узла № 3 СУБД PostgreSQL для БД приложения DHRA при установке в кластере.
17	webiomed_rabbitmq_1=true	да	Узел для размещения RabbitMQ. При установке в кластере узел №1 кластера
18	webiomed_rabbitmq_2=true	да	Узел для размещения узла №2 кластера RabbitMQ.
19	webiomed_rabbitmq_3=true	да	Узел для размещения узла №3 кластера RabbitMQ.
20	webiomed_redis_1=true	да	Узел для размещения Redis. При установке в кластере узел №1 кластера.
21	webiomed_redis_2=true	да	Узел для размещения узла №2 кластера Redis.
22	webiomed_redis_3=true	да	Узел для размещения узла №3 кластера Redis.
23	webiomed_zookeeper_1=true	да	Узел для размещения Zookeeper. При установке в кластере узел №1 кластера.
24	webiomed_zookeeper_2=true	да	Узел для размещения узла №2 кластера Zookeeper.
25	webiomed_zookeeper_3=true	да	Узел для размещения узла №3 кластера Zookeeper.

Таблица 9. Метки инструментов мониторинга.

№	Название метки	Уникальная (только на одном узле)	Назначение
1	monitoring=true	да	Узел для размещения приложений мониторинга

Име. № подл.	Подпись и дата
Взам. инв. №	Име. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

10.4 Описание параметров файлов настроек

10.4.1 Описание параметров системных настроек

Таблица 10. Системные параметры, устанавливаемые в файле /etc/webiomed/system.conf

№	Параметр	Значение по умолчанию	Описание
1	SWARM_ENABLED	-	Определяет используемый оркестратор для работы платформы. При true – Docker Swarm, при false – Docker-compose
2	HA_ENABLED	false	При true приложения Apache Kafka, RabbitMQ и Redis устанавливаются в кластере, при false устанавливается только по одному экземпляру приложений.
3	BIGDATA_ENABLE D	false	При true для установки БД приложений Dataset и DHRA используются отдельные экземпляры СУБД. При false все БД устанавливаются на один сервера СУБД
4	REGISTRY_HOST	-	Адрес локального реестра образов. При использовании встроенного в платформу реестра следует использовать адрес localhost:5000/ (обязательно с символом «/» в конце адреса!).
5	USE_TZ	false	Определяет использование временной зоны из настроек. При true приложения используют время в указанной временной зоне.
6	TIME_ZONE	Europe/Moscow	Используемая временная зона, при USE_TZ=true.
7	PRODUCTION	false	Определяет тип установки (промышленная/тестовая), при true – промышленная,

Ине. № подл.	Подпись и дата
Взам. инв. №	Ине. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					45

			false – тестовая. Используется для вывода соответствующей информации в интерфейсе платформы.
8	DEBUG	false	При true включается отладочный режим работы приложений.
9	SENTRY	false	Включает отправку диагностических данных в сервис Sentry.
10	SENTRY_DSN	-	Указывается DSN Sentry при использовании отладочных сообщений (при SENTRY=true)
11	MONITORING	true	При true выполняется сбор метрик работы приложениями Webiomed в формате Prometheus. При false сбор и публикация метрик не выполняется.
12	BASE_URL	http://127.0.0.1:8000	Адрес платформы. Данные адрес используется при формировании некоторых ссылок приложениями.
13	FRONTEND_V1_HOST	-	Используется для совместимости с версией 1.0 платформы. Следует установить такой же адрес как и в BASE_URL.
14	FRONTEND_V2_HOST	-	Используется для совместимости с версией 1.0 платформы. Следует установить такой же адрес как и в BASE_URL.
15	DATASET_BASE_URL	-	Используется для совместимости с версией 1.0 платформы. Следует установить такой же адрес как и в BASE_URL.

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата

10.4.2 Описание параметров приложения Dataset

Таблица 11. Параметры приложения Dataset, устанавливаемых в файле /etc/secrets/dataset.conf.

№	Параметр	Значение по умолчанию	Описание
1	DATASET_POSTGRES_PASSWORD	-	Пароль для подключения к базе данных dataset. Генерируется автоматически при установке, если не установлен самостоятельно.
2	DATASET_SECRET_KEY	-	Ключ, используемый для обеспечения безопасности API. Генерируется автоматически при установке, если не установлен самостоятельно/

10.4.3 Описание параметров приложения DHRA

№	Параметр	Значение по умолчанию	Описание
1	DHRA_POSTGRES_PASSWORD	-	Пароль для подключения к базе данных dhra. Генерируется автоматически при установке, если не установлен самостоятельно.
2	DHRA_SECRET_KEY	-	Ключ, используемый для обеспечения безопасности API. Генерируется автоматически при установке, если не установлен самостоятельно/

10.4.4 Описание параметров приложения NLP

№	Параметр	Значение по умолчанию	Описание
1	NLP_POSTGRES_PASSWORD	-	Пароль для подключения к базе данных nlp. Генерируется автоматически при установке, если не установлен самостоятельно.

Име. № подл.	
Подпись и дата	
Взам. инв. №	
Име. № дубл.	
Подпись и дата	

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					47

2	NLP_SECRET_KEY	-	Ключ, используемый для обеспечения безопасности API. Генерируется автоматически при установке, если не установлен самостоятельно/
---	----------------	---	---

10.4.5 Описание параметров приложения Symptom-checker

№	Параметр	Значение по умолчанию	Описание
1	SYMPTOM_CHECKER_POSTGRES_PASSWORD	-	Пароль для подключения к базе данных symptom-checker. Генерируется автоматически при установке, если не установлен самостоятельно.
2	SYMPTOM_CHECKER_SECRET_KEY	-	Ключ, используемый для обеспечения безопасности API. Генерируется автоматически при установке, если не установлен самостоятельно/

10.4.6 Описание параметров приложения PatientID

№	Параметр	Значение по умолчанию	Описание
1	PATIENTID_POSTGRES_PASSWORD	-	Пароль для подключения к базе данных patientid. Генерируется автоматически при установке, если не установлен самостоятельно.
2	PATIENTID_SECRET_KEY	-	Ключ, используемый для обеспечения безопасности API. Генерируется автоматически при установке, если не установлен самостоятельно/

Име. № подл.	Подпись и дата
Взам. инв. №	Име. № дубл.
Подпись и дата	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					48

10.4.7 Описание параметров приложения Gate

№	Параметр	Значение по умолчанию	Описание
1	GATE_POSTGRES_PASSWORD	-	Пароль для подключения к базе данных gate. Генерируется автоматически при установке, если не установлен самостоятельно.
2	GATE_SECRET_KEY	-	Ключ, используемый для обеспечения безопасности API. Генерируется автоматически при установке, если не установлен самостоятельно/

10.4.8 Описание параметров приложения DMA

№	Параметр	Значение по умолчанию	Описание
1	DMA_POSTGRES_PASSWORD	-	Пароль для подключения к базе данных dma. Генерируется автоматически при установке, если не установлен самостоятельно.
2	DMA_SECRET_KEY	-	Ключ, используемый для обеспечения безопасности API. Генерируется автоматически при установке, если не установлен самостоятельно/

Име. № подл.	Подпись и дата	Взам. инв. №	Име. № дубл.	Подпись и дата

Изм.	Лист	№ докум.	Подпись	Дата	Лист
					49