

АО «РОТЕК Диджитал Солюшнс»

«УТВЕРЖДАЮ»
Генеральный директор

А.В. Куканов

Инструкция по установке программного обеспечения
**«Система Прогностики и Удаленного Мониторинга
(СПиУМ) «ПРАНА»»**

Москва, 2022

© АО «РОТЕК Диджитал Солюшнс»

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

Разраб.				
---------	--	--	--	--

Пров.				
-------	--	--	--	--

Н. контр.				
-----------	--	--	--	--

Утв.				
------	--	--	--	--

Инструкция по установке ПО
«Система Прогностики и
Удаленного Мониторинга
(СПиУМ) «ПРАНА»»

Лит.	Лист	Листов
------	------	--------

	1	
--	---	--

--	--	--

Перв. Примен.	ОГЛАВЛЕНИЕ					
	ТЕРМИНОЛОГИЯ.....					2
Справ. №	ВВЕДЕНИЕ					2
	1. СОСТАВ УСТАНОВОЧНОГО КОМПЛЕКТА					3
	1.1. Требования к оборудованию серверов.....					3
	1.2. Требования к ОС серверов.					3
	2. УСТАНОВКА ПО ПРАНА.....					4
	2.1. Копирование установочных файлов					4
	2.2. Создание директорий на сервере установки (Сервере приложений)					5
	2.2.1. Директория web.....					6
	2.2.2. Директория go_cont.					15
	2.2.3. Директория WinShare.....					15
	2.2.4. Директория demo-schema.....					15
	2.3. Перенос контейнеров					15
	2.3.1. Архивирование и копирование контейнеров на сервер:					15
	2.3.2. Разворачивание архива.....					16
	2.4. Создание баз данных.					16
	2.4.1. Создание БД.....					16
	Запускаем команду CREATE DATABASE PranaRAW					16
	2.4.2. Создание таблиц.....					16
	2.4.3. Создание материализованных представлений.....					17
	2.4.4. Перенос основных схем.					19
	2.5. Копирование и настройка переменных для файлов Docker Compose.					21
	2.5.1. Листинг файла docker-compose.yml					21
	2.5.2. Листинг compose-api-gw-master.yml					23
	2.5.3. Листинг compose-clickhouse_http_api-master.yml.....					24
	2.5.4. Листинг compose-collector_cfg_distributor-master.yml					25
	2.5.5. Листинг compose-data_delay_detector-master.yml.....					25
	2.5.6. Листинг compose-equipment-type-go-master.yml					26
	2.5.7. Листинг compose-event-writer-master.yml					27
	2.5.8. Листинг compose-go-analytics-master.yml					28
Инв. № подл						
Подп. и дата						
Взам. инв. №						
Инв. № дубл						
Подп. и дата						
Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально					<i>Лист</i>	
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>	2	

Инв.№ подл	Подп. и дата	Взам.инв.№	Инв.№ дубл	Подп. и дата	Справ.№	Перв. Примен.	2.5.9. Листинг compose-go-doc_storage_api-master.yml 29
							2.5.10. Листинг compose-go-unit_nodes_api-master.yml 29
							2.5.11. Листинг compose-kafka_slicer-master.yml 30
							2.5.12. Листинг compose-kh_writer-master.yml..... 30
							2.5.13. Листинг compose-param-threshold-master.yml 31
							2.6. Команды запуска.32
Изм	Лист	№ докум.	Подп.	Дата			Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально

Перв. Примен.	
Справ. №	

ТЕРМИНОЛОГИЯ

Термин	Пояснения
ПО ПРАНА	Программное обеспечение «Система Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА»»
БД	Базы данных
Пользователь	Пользователь «Системы Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА»»
Docker	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер
Папка. Директория.	Папки (или директории) выделены фиолетовым цветом (размером 11 pt)
Файлы	Файлы выделены черным цветом (размером 11 pt)
Текст файла	Текст файла (листинг) выделен черным цветом (размером 9 pt)
Дистрибутив для установки	Предоставляемый Набор файлов, который необходим для разворачивания ПО «Система Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА»».

ВВЕДЕНИЕ

ПО «Система Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА»» представляет собой достаточно сложную программу, которая является частью программно-аппаратного комплекса. Процесс его установки занимает не менее одной недели.

Установку ПО «Система Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА»» может произвести только специалист со знаниями вопросов установки программного обеспечения на базе контейнеризации.

Настоятельно рекомендуется поручить установку ПО «Система Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА»» специалистам компании АО «РОТЕК Диджитал Солюшнс».

ПО «Система Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА»» предназначена для диагностики и прогнозирования объектов энергетики, а также для автоматизированного выявления дефектов на объектах на начальной стадии их развития. Это, в конечном итоге, позволяет сэкономить значительные средства, которые потенциально могут быть затрачены на устранение последствий аварий. Под объектами понимается сложное технологическое оборудование. Например.

- Газовые турбины.
- Паровые турбины.
- Генераторы.
- Трансформаторы.
- Котлы и сопутствующее оборудование.
- Дожимные компрессоры.
- Ветроэнергоустановки.

В принципе возможно использование Системы ПРАНА на любом оборудовании.

Способ приобретения программного обеспечения: определяется в индивидуальном порядке для каждого клиента.

Информация о местах нахождения (фактические адреса):

Фактический адрес размещения инфраструктуры разработки:
РФ, 109240, г. Москва, ул. Николаямская, д.15, оф. 105.

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

Изм	Лист	№ докум.	Подп.	Дата
-----	------	----------	-------	------

Перв. Примен.	<p>Фактический адрес размещения службы поддержки: РФ, 109240, г. Москва, ул. Николаямская, д.15, оф. 105.</p> <p>Фактический адрес размещения серверов: РФ, 109240, г. Москва, ул. Николаямская, д.15</p> <p>Фактический адрес размещения разработчиков: РФ, 109240, г. Москва, ул. Николаямская, д.15, оф. 105.</p>					
	Справ. №	<p>1. СОСТАВ УСТАНОВОЧНОГО КОМПЛЕКТА</p> <p>Для нормальной работы ПО «Система Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА»» рекомендуется иметь три сервера. Перед установкой необходимо убедиться, что серверы, на которых будет производится установка удовлетворяют следующим требованиям.</p> <p>1.1. Требования к оборудованию серверов</p> <p>Сервер потокового вещания данных (Kafka) Кол-во ядер процессора 8 RAM 8Гб Дисковое хранилище высокопроизводительное 300ГБ</p> <p>Сервер СУБД ClickHouse Кол-во ядер процессора 12 RAM 32Гб Дисковое хранилище HDD 1ТБ</p> <p>Сервер приложений Кол-во ядер процессора 12 RAM 24Гб Дисковое хранилище высокопроизводительное 500ГБ</p>				
Подп. и дата		<p>1.2. Требования к ОС серверов.</p> <p>На серверах может быть использован любой Linux дистрибутив. В данном описании инструкции даны для Ubuntu 18.04.5 LTS.</p> <p>Перед установкой нужно убедиться, что на серверах) установлены следующие программы:</p> <p>На сервере Сервер потокового вещания данных Kafka На сервере приложений Docker и Docker-compose, На сервер СУБД ClickHouse - ClickHouse, PostgreSQL. Наличие данных программ является обязательным.</p>				
	Инв. № дубл					
Взам. инв. №						
	Подп. и дата					
Инв. № подл						
	Изм	Лист	№ докум.	Подп.	Дата	<p>Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>
					3	

Перв. Примен.	<h2 style="text-align: center;">2. УСТАНОВКА ПО ПРАНА</h2> <p>Как уже было указано, установка является крайне сложным процессом. Если по каким-либо причинам самостоятельная установка не будет получаться, для целей демонстрации работы ПО «Система Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА»» развернут специальный стенд, в котором можно посмотреть функциональность ПО.</p> <p>Данные для входа на стенд. Адрес: <code>http://89.249.55.87:20080</code> Login: <code>expert</code> Password: <code>expert_pass</code></p>																						
Справ. №	<p>Краткое описание процесса. Процесс установки представляет собой следующие шаги.</p> <ul style="list-style-type: none"> • Подготовка серверов (установка на них необходимых программ (см. п. 1.2.), создание рабочих директорий (см. п.2.1.). • Копирование установочных файлов (которые представляют собой шесть архивов) в определенные директории на серверах. • Разархивирование скопированных архивов. • загрузка docker образов из архивов. • Создание баз данных. • Копирование и настройка переменных для файлов Docker Compose. • Запуск <p>Далее каждый пункт будет рассмотрен более подробно.</p> <h3 style="text-align: center;">2.1. Копирование установочных файлов</h3> <p>Эксперт может получить доступ к экземпляру программного обеспечения, выполнив следующие действия:</p> <ol style="list-style-type: none"> 1. Доступ с помощью SSH, команда вида: <code>ssh expert_demo@89.249.55.87:50550</code> <code>cd /mnt/distr-resources</code> путь к папке, в которой будут лежать файлы дистрибутива <code>/mnt/distr-resources</code> Login <code>expert_demo</code> Password <code>bb36bc541fd983d3827f0249</code> <p>2. После успешной авторизации на сервере открывается доступ к модулям экземпляра программного обеспечения. Компоненты расположены на этом же сервере. Ниже в описании компонентов для каждого компонента приводится его назначение.</p> <p>Установочные файлы представляют собой шесть архивов (см.рис. ниже)</p> <table border="1" data-bbox="188 1630 1497 1848"> <tr> <td><code>WinShare.tar</code></td> <td><code>1664M</code></td> <td><code>Jan 30 13:33</code></td> </tr> <tr> <td><code>*demo-schema.rar</code></td> <td><code>376704K</code></td> <td><code>Jan 30 13:54</code></td> </tr> <tr> <td><code>go_cont.tar</code></td> <td><code>30720</code></td> <td><code>Jan 30 13:32</code></td> </tr> <tr> <td><code>go_docker.tar.gz</code></td> <td><code>9944M</code></td> <td><code>Jan 30 13:52</code></td> </tr> <tr> <td><code>web.tar</code></td> <td><code>2171M</code></td> <td><code>Jan 30 13:35</code></td> </tr> <tr> <td><code>web_docker.tar.gz</code></td> <td><code>3590M</code></td> <td><code>Jan 30 13:40</code></td> </tr> </table> <ol style="list-style-type: none"> 1.WinShare –архив с моделями. 2.Demo-schema – архив с дампами схем для развертывания в БД. Переносится в место, в котором будет производится настройка схем БД PostgreSQL. То есть на сервер СУБД ClickHouse. 3.Go_cont – архив с конфигурационными файлами. 					<code>WinShare.tar</code>	<code>1664M</code>	<code>Jan 30 13:33</code>	<code>*demo-schema.rar</code>	<code>376704K</code>	<code>Jan 30 13:54</code>	<code>go_cont.tar</code>	<code>30720</code>	<code>Jan 30 13:32</code>	<code>go_docker.tar.gz</code>	<code>9944M</code>	<code>Jan 30 13:52</code>	<code>web.tar</code>	<code>2171M</code>	<code>Jan 30 13:35</code>	<code>web_docker.tar.gz</code>	<code>3590M</code>	<code>Jan 30 13:40</code>
<code>WinShare.tar</code>	<code>1664M</code>	<code>Jan 30 13:33</code>																					
<code>*demo-schema.rar</code>	<code>376704K</code>	<code>Jan 30 13:54</code>																					
<code>go_cont.tar</code>	<code>30720</code>	<code>Jan 30 13:32</code>																					
<code>go_docker.tar.gz</code>	<code>9944M</code>	<code>Jan 30 13:52</code>																					
<code>web.tar</code>	<code>2171M</code>	<code>Jan 30 13:35</code>																					
<code>web_docker.tar.gz</code>	<code>3590M</code>	<code>Jan 30 13:40</code>																					
Подп. и дата																							
Име.№ дубл																							
Взам.име.№																							
Подп. и дата																							
Име.№ подл																							
Изм	Лист	№ докум.	Подп.	Дата	Инструкция по установке © АО «ПОТЕК Диджитал Солюшнс» Конфиденциально	Лист 4																	

Перв. Примен.

Справ. №

- Переносится на сервер приложений.
- 4.Go_docker – архив с образами контейнеров go.
Переносится на сервер приложений.
- 5.Web – архив с файлами проекта и конфигурационными файлами web сервисов.
Переносится на сервер приложений
- 6.Web_docker – архив с всеми контейнерами web.
Переносится на сервер приложений.

Архивы с пометкой docker разворачиваются командой `docker load < имя_архива.tar`
Архивы с данными приложения разворачиваются командой `tar -xvf ИМЯ_АРХИВА.tar /путь в место, куда разархивируем архив.`

Распаковав архивы – получим загруженные образы и compose-файлы.
В конце процесса установки (см. раздел 2.6) compose-файлы нужно будет запустить. В результате будет создано 19 docker-контейнеров.

№№	Название Docker-контейнера	Пояснения
1	kh_writer:master	Сервис, который забирает данные из очереди kafka и транслирующий их в БД Clickhouse
2	go-analytics:master	Сервис расчета аналитики на основе потока входящих измерений
3	collector_cfg_distributor:master	Сервис конфигураций
4	kafka_slicer:master	Сервис обеспечивает создание «срезов» состояния данных. Т.е. раз в выбранный интервал присылает последние значения измерений (например –для станции)
5	nginx:1.17	Веб сервер
6	clickhouse_http_api:master	API для доступа к данным измерений в ClickHouse
7	param-threshold:master	Сервис обеспечивает проверку выхода параметров за назначенные им пределы (так называемые уставки)
8	event-writer:master	Сервис записи событий по данным параметров со станции
9	equipment-type-go:master	Каталог оборудования
10	go-unit_nodes_api:master	Каталог узлов агрегатов. Каталог дефектов агрегатов. API автоматических правил
11	go-doc_storage_api:master	НСИ (нормативно-справочная информация) сервис каталога документов
12	data_delay_detector:master	Сервис обнаружения задержки поступления данных в систему
13	api-gw:master	Шлюз API сервисов go
14	web_php	API доступа для взаимодействия с клиентским приложением
15	system_system	Сервис с набором служб для работы системы
16	web_node	Фронт (Frontend)
17	redis:alpine	NOsql БД для кэширования данных
18	rabbitmq:3-management	Брокер сообщений, для обработки событий
19	centrifugo/centrifugo:v2.2.6	Сервис вебсокетов (frontend websockets).

2.2. Создание директорий на сервере установки (Сервере приложений)

Для разворачивания системы на Сервере приложений создаем рабочий каталог (например: в /mnt или home- домашнем каталоге пользователя).

Далее распаковываем в рабочий каталог поочередно архивы (команды описаны в разделе 2.1.)
В результате получаем набор каталогов.

- web

Подп. и дата

Инв.№ дубл

Взам. инв.№

Подп. и дата

Инв.№ подл

- go_cont
- WinShare
- demo-schema

2.2.1. Директория web.

Должна содержать в себе следующие папки и файлы (нумерация дана для удобства):

1	centrifugo	/centrifugo
2	custom_log	/custom_log
3	nginx	/nginx
4	prana.api	/prana.api
5	prana.client	/prana.client
6	rabbitmq	/rabbitmq
7	redis	/redis
8	relevance (не используется)	/relevance
9	resources	/resources
10	system	/system
11	.docker-compose.yml	*.dockerignore *.gitlab-ci.yml *docker-compose.yml

Содержимое папки (1) centrifuge:

Файл config.json

Конфигурация файла config.json

```
{
  "secret": "secret",
  "api_key": "key",
  "admin": true,
  "admin_password": "XXX",
  "admin_secret": "XXX",
  "web": true,
  "port": "8000",
  "log_level": "debug",
  "connection_lifetime": 0,
  "anonymous": true,
  "client_insecure": true,
  "publish": true,
  "broadcast": true,
  "history": true,
  "unsubscribe": true,
  "disconnect": true,
  "channels": true,
  "info": true,
  "subscribe_to_publish": true,
  "presence": true,
  "join_leave": true,
  "history_recover": true,
  "presence_stats": true,
```


Перв. Примен.		<pre> "watch": true, "debug": true, "history_size": 5, "history_lifetime": 100, "prometheus": true } </pre> <p>Содержимое папки (2) custom_log.</p> <p>Папка, которая монтируется в контейнер nginx и также в контейнер php. Предназначена для логов и сервисов web.</p>													
Справ. №		<p>Содержимое папки (3) nginx:</p> <p>Файл nginx.conf</p> <p>Конфигурация файла nginx.conf</p> <pre> worker_processes 1; events { worker_connections 1024; } http { keepalive_requests 1000; keepalive_timeout 30; log_format compression '\$remote_addr - \$remote_user [\$time_local] ' '\$request' \$status \$bytes_sent ' '\$http_referer' '\$http_user_agent' '\$gzip_ratio'; access_log /var/log/nginx/access.log compression; error_log /var/log/nginx/error.log warn; reset_timedout_connection on; client_body_timeout 480; client_max_body_size 300m; send_timeout 480; x server { listen 80; server_name localhost; root /usr/share/nginx/html; #client location / { proxy_pass http://node:81; </pre>													
Подп. и дата															
Инв. № дубл															
Взам. инв. №															
Подп. и дата															
Инв. № подл		<table border="1"> <tr> <td data-bbox="135 2069 193 2145"></td> <td data-bbox="193 2069 266 2145"></td> <td data-bbox="266 2069 440 2145"></td> <td data-bbox="440 2069 555 2145"></td> <td data-bbox="555 2069 628 2145"></td> <td data-bbox="628 2069 1465 2145" rowspan="2" style="text-align: center; vertical-align: middle;"> Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально </td> <td data-bbox="1465 2069 1541 2145" style="text-align: center;">Лист</td> </tr> <tr> <td data-bbox="135 2145 193 2181">Изм</td> <td data-bbox="193 2145 266 2181">Лист</td> <td data-bbox="266 2145 440 2181">№ докум.</td> <td data-bbox="440 2145 555 2181">Подп.</td> <td data-bbox="555 2145 628 2181">Дата</td> <td data-bbox="1465 2145 1541 2181" style="text-align: center;">7</td> </tr> </table>						Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально	Лист	Изм	Лист	№ докум.	Подп.	Дата	7
					Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально	Лист									
Изм	Лист	№ докум.	Подп.	Дата		7									

Перв. Примен.	<pre> } location ^~ /cfgman/ { proxy_pass http://collector_cfg_distributor:25111; rewrite '^/cfgman/(.*)\$' /\$1 break; } location ^~ /api/gate/ { proxy_pass http://api-gw:25113; rewrite '^/api/gate/(.*)\$' /\$1 break; } #centrifugo location ^~ /connection/ { proxy_pass http://IP-АДРЕС-СЕРВЕРА:8001; proxy_http_version 1.1; proxy_set_header Upgrade \$http_upgrade; proxy_set_header Connection "upgrade"; proxy_set_header Host \$host; } location ^~ /api/ { add_header Access-Control-Allow-Origin \$http_origin; add_header Access-Control-Allow-Methods 'GET, POST, PUT, DELETE, OPTIONS'; add_header Access-Control-Allow-Headers 'DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Content-Range,Range'; add_header Access-Control-Expose-Headers 'DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Content-Range,Range'; add_header Access-Control-Allow-Credentials 'true'; add_header Vary Origin; rewrite '^/api/(.*)\$' /index.php break; fastcgi_split_path_info ^(/api/)(.*)\$; fastcgi_pass php:9000; fastcgi_index index.php; fastcgi_send_timeout 300; fastcgi_read_timeout 300; fastcgi_param SCRIPT_FILENAME \$document_root\$fastcgi_script_name; include /etc/nginx/fastcgi_params; } } </pre>					
Справ. №						
Подп. и дата						
Инв. № дубл						
Взам. инв. №						
Подп. и дата						
Инв. № подл						
Изм	Лист	№ докум.	Подп.	Дата	<p align="center">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>	<p align="right">Лист 8</p>

Ине.№ подл	Подп. и дата	Взам. ине.№	Ине.№ дубл	Подп. и дата	Справ.№	Перв. Примен.	<p>Содержимое папки (4) prana.api</p> <pre> .env .htaccess .php_cs Dockerfile app builder_dev.sh builder_prod.sh cert composer-setup.php composer.json composer.lock composer.phar config_main_postgres_conn.ini config_temp_postgres_conn.ini db db2 db_unification docs favicon.ico general_config.ini index.php logs phinx.php report.html vendor </pre> <hr/> <p>Файл конфигурации .env</p> <pre> # CENTRIFUGO CENTRIFUGO_IP="ip" CENTRIFUGO_PORT="8001" CENTRIFUGO_API_KEY="key" CENTRIFUGO_SECRET_KEY="secret" # RABBIT RABBIT_IP="ip" RABBIT_PORT="5673" RABBIT_MODE="release" RABBIT_SERVER="NN" RABBIT_USER="rabbitmq" RABBIT_PASS="rabbitmq" # REDIS REDIS_CONFIG_SCHEME="tcp" </pre>
							<p>Изм</p>
<p>Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>						<p>Лист</p>	
						<p>9</p>	

Перв. Примен.		<pre> REDIS_CONFIG_HOST="redis" REDIS_CONFIG_PORT="6379" # ClickHouse CH_HOST="ip" CH_PORT="9000" CH_USERMAME="default" </pre>			
Справ. №		<pre> # MAIN POSTGERS CONNECTS #- Стабильные адреса БД, копировать на каждый рабочий проект DB1_HOST="ip" DB1_PORT="5432" DB1_USER="user" DB1_PASSWORD="Pass " DB1_DBNAME="postgres" MSET_DB_HOST="ip" MSET_DB_PORT="5432" MSET_DB_USER="user" MSET_DB_PASSWORD="Pass" MSET_DB_DBNAME="postgres" </pre>			
Подп. и дата		<pre> USER_DB_HOST="ip" USER_DB_PORT="5432" USER_DB_USER="user" USER_DB_PASSWORD="Pass" USER_DB_DBNAME="postgres" </pre>			
Инв. № дубл		<pre> DB2_HOST="ip" DB2_PORT="5432" DB2_USER="user" DB2_PASSWORD="Pass" DB2_DBNAME="postgres" </pre>			
Взам. инв. №		<pre> LOG_DB_HOST="ip" LOG_DB_PORT="5432" LOG_DB_USER="user" LOG_DB_PASSWORD="Pass" LOG_DB_DBNAME="postgres" </pre>			
Подп. и дата		<pre> AI_DB_HOST="ip" AI_DB_PORT="5432" AI_DB_USER="user" </pre>			
Инв. № подл					
Изм	Лист	№ докум.	Подп.	<p align="center">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>	<p align="right">Лист 10</p>

Перв. Примен.		<pre> AI_DB_PASSWORD="Pass" AI_DB_DBNAME="postgres" # END - MAIN POSTGERS CONNECTS GOCH_HOST="ip:7890" # Локальные системные пути для API resources_local="/var/www/resources/" log_folder="/var/log/prana.api/custom_log/" service_log_folder="/mnt/data/web/custom_log/prana.api/" # Внешние системные пути для API, только чтение models="/var/www/resources/station_models/" raw_data_9b="/var/www/resources/data/raw/" an_data_9b="/var/www/resources/data/an/" # Системные пути, для PYMSET dataset_folder="/var/www/resources/pymset/datasets/" models_folder="/var/www/resources/pymset/models/" stations_folder="/var/www/resources/pymset/station_models/" # Внешние системные путь для ресурсов resources="/var/www/resources/" # Системный путь для выгрузок этого сервера export_folder="/var/www/resources/export/" # Адрес выгрузок для юзера из этого сервера # export_folder_user="/Prana.Api_dev/export/" export_folder_user="/var/www/resources/export/" # App remote_ip = "REMOTE_ADDR" python_server = "pyserver:9080" # Constants log_bin_time=0 max_data_delay=300000 </pre>																							
Справ. №																									
Подп. и дата																									
Инв. № дубл																									
Взам. инв. №																									
Подп. и дата		<p>Содержимое папки (5) prana.client</p> <pre> client.json package-lock.json public server.js </pre>																							
Инв. № подл		<table border="1"> <tr> <td data-bbox="135 2072 199 2116"></td> <td data-bbox="199 2072 263 2116"></td> <td data-bbox="263 2072 327 2116"></td> <td data-bbox="327 2072 391 2116"></td> <td data-bbox="391 2072 454 2116"></td> <td data-bbox="454 2072 518 2116"></td> <td data-bbox="518 2072 582 2116"></td> <td data-bbox="582 2072 646 2116"></td> <td data-bbox="646 2072 710 2116"></td> <td data-bbox="710 2072 774 2116"></td> <td data-bbox="774 2072 1460 2116" rowspan="2"> <p align="center">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p> </td> <td data-bbox="1460 2072 1540 2116">Лист</td> </tr> <tr> <td data-bbox="135 2116 199 2184">Изм</td> <td data-bbox="199 2116 263 2184">Лист</td> <td data-bbox="263 2116 327 2184">№ докум.</td> <td data-bbox="327 2116 391 2184">Подп.</td> <td data-bbox="391 2116 454 2184">Дата</td> <td data-bbox="454 2116 518 2184"></td> <td data-bbox="518 2116 582 2184"></td> <td data-bbox="582 2116 646 2184"></td> <td data-bbox="646 2116 710 2184"></td> <td data-bbox="710 2116 774 2184"></td> <td data-bbox="1460 2116 1540 2184">11</td> </tr> </table>											<p align="center">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>	Лист	Изм	Лист	№ докум.	Подп.	Дата						11
										<p align="center">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>	Лист														
Изм	Лист	№ докум.	Подп.	Дата							11														

Перв. Примен.		<p>Конфигурация файла server.js</p> <pre> const express = require('express'); const path = require('path'); const app = express(); app.get('*.*', function (req, res, next){ if (!['service-worker', 'precache', 'workbox'].every(v => !req.url.includes(v))){ req.url = req.url + '.gz'; res.set('Content-Encoding', 'gzip'); res.set('Content-Type', 'text/javascript'); } next(); }); app.use(express.static(path.join(__dirname, 'public'))); app.get('/ping', function (req, res) { return res.send('pong'); }); app.get('/', function (req, res) { res.sendFile(path.join(__dirname, 'public', 'index.html')); }); app.get('/*', function(req, res) { res.sendFile(path.join(__dirname, 'public', 'index.html')); }); app.listen(81, '0.0.0.0', (err) => { if (err){ throw new Error(err); } console.log('Listening on port 81'); }); </pre>																							
Справ. №																									
Подп. и дата																									
Инв. № дубл																									
Взам. инв. №																									
Подп. и дата		<p>Содержимое папки (6) rabbitmq</p> <pre> definitions.json enabled_plugins rabbitmq.config </pre> <p>Конфигурация файла rabbitmq.config</p> <pre> [{rabbit, [</pre>																							
Инв. № подл		<table border="1"> <tr> <td data-bbox="135 2069 193 2123"></td> <td data-bbox="193 2069 266 2123"></td> <td data-bbox="266 2069 339 2123"></td> <td data-bbox="339 2069 413 2123"></td> <td data-bbox="413 2069 486 2123"></td> <td data-bbox="486 2069 560 2123"></td> <td data-bbox="560 2069 633 2123"></td> <td data-bbox="633 2069 707 2123"></td> <td data-bbox="707 2069 780 2123"></td> <td data-bbox="780 2069 853 2123"></td> <td data-bbox="853 2069 1465 2123" rowspan="2">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</td> <td data-bbox="1465 2069 1541 2123">Лист</td> </tr> <tr> <td data-bbox="135 2123 193 2181">Изм</td> <td data-bbox="193 2123 266 2181">Лист</td> <td data-bbox="266 2123 339 2181">№ докум.</td> <td data-bbox="339 2123 413 2181">Подп.</td> <td data-bbox="413 2123 486 2181">Дата</td> <td data-bbox="486 2123 560 2181"></td> <td data-bbox="560 2123 633 2181"></td> <td data-bbox="633 2123 707 2181"></td> <td data-bbox="707 2123 780 2181"></td> <td data-bbox="780 2123 853 2181"></td> <td data-bbox="1465 2123 1541 2181">12</td> </tr> </table>											Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально	Лист	Изм	Лист	№ докум.	Подп.	Дата						12
										Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально	Лист														
Изм	Лист	№ докум.	Подп.	Дата							12														

Перв. Примен.		<pre> (loopback_users, []), {vm_memory_high_watermark, 0.7}, {vm_memory_high_watermark_paging_ratio, 0.8}, {log_levels, [{channel, warning}, {connection, warning}, {federation, warning}, {mirroring, info}]}, {heartbeat, 10}]}, {rabbitmq_management, [{load_definitions, "/etc/rabbitmq/definitions.json"}]}]. </pre>				
Справ. №		<p>Содержимое папки (7) redis</p> <pre> redis.conf Файл конфигурации redis.conf # Redis configuration file example. bind 0.0.0.0 protected-mode no port 6378 tcp-backlog 511 timeout 0 tcp-keepalive 300 daemonize yes supervised no pidfile /var/run/redis/redis-server.pid loglevel notice logfile /var/log/redis/redis-server.log databases 16 always-show-logo yes save 900 1 save 300 10 save 60 10000 stop-writes-on-bgsave-error yes rdbcompression yes rdbchecksum yes dbfilename dump.rdb dir /var/lib/redis slave-serve-stale-data yes slave-read-only yes repl-diskless-sync no repl-diskless-sync-delay 5 </pre>				
Подп. и дата						
Инв. № дубл						
Взам. инв. №						
Подп. и дата						
Инв. № подл						
Изм	Лист	№ докум.	Подп.	Дата	<p align="center">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>	Лист
						13

Перв. Примен.				
Справ. №		repl-disable-tcp-nodelay no slave-priority 100 lazyfree-lazy-eviction no lazyfree-lazy-expire no lazyfree-lazy-server-del no slave-lazy-flush no appendonly no appendfilename "appendonly.aof" appendfsync everysec no-appendfsync-on-rewrite no auto-aof-rewrite-percentage 100 auto-aof-rewrite-min-size 64mb aof-load-truncated yes aof-use-rdb-preamble no lua-time-limit 5000 slowlog-log-slower-than 10000 slowlog-max-len 128 latency-monitor-threshold 0 notify-keyspace-events "" hash-max-ziplist-entries 512 hash-max-ziplist-value 64 list-max-ziplist-size -2 list-compress-depth 0 set-max-intset-entries 512 zset-max-ziplist-entries 128 zset-max-ziplist-value 64 hll-sparse-max-bytes 3000 activerehashing yes client-output-buffer-limit normal 0 0 0 client-output-buffer-limit slave 256mb 64mb 60 client-output-buffer-limit pubsub 32mb 8mb 60 hz 10 aof-rewrite-incremental-fsync yes		
Подп. и дата				
Инв. № дубл				
Взам. инв. №		<p>Содержимое папки (8) resources</p> <p>Предназначена для загрузки данных в систему. Хранения тегов, мнемосхем, картинок.</p>		
Подп. и дата		<p>Содержимое папки (10) system</p> <p>docker-compose.yml -user.target.wants websocket_access_ws.service websocket_analytics_ws.service websocket_analytics_ws.timer websocket_event_journal_ws.service</p>		
Инв. № подл				
Изм	Лист	№ докум.	Подп.	Дата
Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально				Лист 14

Перв. Примен.	<p>websocket_events_ws.service</p> <p>websocket_main_alert_ws.service</p> <p>В вышеуказанных папках находятся:</p> <ul style="list-style-type: none"> - конфигурации сервисов запущенных в контейнерах отвечающих за веб составляющую проекта, - логи (Logs) системы, - папки для загрузки данных в систему(resources), - а так же непосредственно файл запуска контейнеров docker-compose.yml в котором описаны связи контейнеров, подсети и монтируемые тома (volumes). Перед запуском обязательно проверяются файлы конфигураций всех сервисов на правильные адреса БД и связанных систем. 																														
	Справ. №	<p>2.2.2. Директория go_cont.</p> <p>Содержит в себе файлы конфигурации\запуска контейнеров сервисов, переписанных на языке Go. Каждый файл перед запуском проверяется на соответствие конфигурации (адреса БД, адреса брокеров сообщений, адреса kafka, токены авторизаций в системе).</p>																													
Подп. и дата		<table border="1"> <tr> <td>compose-api-gw-master.yml</td> <td>*compose-api-gw-master.yml</td> </tr> <tr> <td>compose-clickhouse_http_api-master.yml</td> <td>*compose-clickhouse_http_api-master.yml</td> </tr> <tr> <td>compose-collector_cfg_distributor-master.yml</td> <td>*compose-collector_cfg_distributor-master.yml</td> </tr> <tr> <td>compose-data_delay_detector-master.yml</td> <td>*compose-data_delay_detector-master.yml</td> </tr> <tr> <td>compose-equipment-type-go-master.yml</td> <td>*compose-equipment-type-go-master.yml</td> </tr> <tr> <td>compose-event-writer-master.yml</td> <td>*compose-event-writer-master.yml</td> </tr> <tr> <td>compose-go-analytics-master.yml</td> <td>*compose-go-analytics-master.yml</td> </tr> <tr> <td>compose-go-doc_storage_api-master.yml</td> <td>*compose-go-doc_storage_api-master.yml</td> </tr> <tr> <td>compose-go-unit_nodes_api-master.yml</td> <td>*compose-go-unit_nodes_api-master.yml</td> </tr> <tr> <td>compose-kafka_slicer-master.yml</td> <td>*compose-kafka_slicer-master.yml</td> </tr> <tr> <td>compose-kh_writer-master.yml</td> <td>*compose-kh_writer-master.yml</td> </tr> <tr> <td>compose-oauth_server-master.yml</td> <td>*compose-oauth_server-master.yml</td> </tr> <tr> <td>compose-param-threshold-master.yml</td> <td>*compose-param-threshold-master.yml</td> </tr> </table>					compose-api-gw-master.yml	*compose-api-gw-master.yml	compose-clickhouse_http_api-master.yml	*compose-clickhouse_http_api-master.yml	compose-collector_cfg_distributor-master.yml	*compose-collector_cfg_distributor-master.yml	compose-data_delay_detector-master.yml	*compose-data_delay_detector-master.yml	compose-equipment-type-go-master.yml	*compose-equipment-type-go-master.yml	compose-event-writer-master.yml	*compose-event-writer-master.yml	compose-go-analytics-master.yml	*compose-go-analytics-master.yml	compose-go-doc_storage_api-master.yml	*compose-go-doc_storage_api-master.yml	compose-go-unit_nodes_api-master.yml	*compose-go-unit_nodes_api-master.yml	compose-kafka_slicer-master.yml	*compose-kafka_slicer-master.yml	compose-kh_writer-master.yml	*compose-kh_writer-master.yml	compose-oauth_server-master.yml	*compose-oauth_server-master.yml	compose-param-threshold-master.yml
	compose-api-gw-master.yml	*compose-api-gw-master.yml																													
compose-clickhouse_http_api-master.yml	*compose-clickhouse_http_api-master.yml																														
compose-collector_cfg_distributor-master.yml	*compose-collector_cfg_distributor-master.yml																														
compose-data_delay_detector-master.yml	*compose-data_delay_detector-master.yml																														
compose-equipment-type-go-master.yml	*compose-equipment-type-go-master.yml																														
compose-event-writer-master.yml	*compose-event-writer-master.yml																														
compose-go-analytics-master.yml	*compose-go-analytics-master.yml																														
compose-go-doc_storage_api-master.yml	*compose-go-doc_storage_api-master.yml																														
compose-go-unit_nodes_api-master.yml	*compose-go-unit_nodes_api-master.yml																														
compose-kafka_slicer-master.yml	*compose-kafka_slicer-master.yml																														
compose-kh_writer-master.yml	*compose-kh_writer-master.yml																														
compose-oauth_server-master.yml	*compose-oauth_server-master.yml																														
compose-param-threshold-master.yml	*compose-param-threshold-master.yml																														
Инв. № дубл	<p>2.2.3. Директория WinShare</p> <p>Содержит так называемые «модели» для системы. Модели предназначены для аналитических расчетов.</p>																														
	Взам. инв. №	<p>2.2.4. Директория demo-schema.</p> <p>Является архивом со схемами Postgres</p>																													
Подп. и дата		<p>2.3. Перенос контейнеров</p>																													
	Инв. № подл	<p>2.3.1. Архивирование и копирование контейнеров на сервер:</p> <p>Перед запуском контейнеров их образы нужно загрузить, используя сохраненные файловые версии контейнеров (прилагаются в дистрибутиве для установки).</p> <p>Образ контейнера архивируется (например, в архив tar) и переносится на сервер, где будет разворачиваться ПО «Система Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА»».</p> <p>Образы архивируются командой 'docker save -o имя_архива.tar образ_который_сохраняете'</p> <p>Пример: docker save -o slicer.tar.gz reg.prana-system.com:443/dev/kafka_slicer:master</p>																													
Изм		Лист	№ докум.	Подп.	Дата																										
<p>Инструкция по установке © АО «РОТЕК Диджитал Солюинс» Конфиденциально</p>					<p>Лист</p> <p>15</p>																										

2.3.2. Разворачивание архива.

После переноса архива с образом на исходный сервер, архив разворачивается командой

```
docker load < имя_архива.tar
```

после этого в списке образов докера появится перенесенный нами образ.

Таким образом в списке образов должны появиться все 19 образов.
Ниже они перечислены.

№№	Название Docker-контейнера	Пояснения
1	kh_writer:master	Сервис, который забирает данные из очереди kafka и транслирующий их в БД Clickhouse
2	go-analytics:master	Сервис расчета аналитики на основе потока входящих измерений
3	collector_cfg_distributor:master	Сервис конфигураций
4	kafka_slicer:master	Сервис обеспечивает создание «срезов» состояния данных. Т.е. раз в выбранный интервал присылает последние значения измерений (например – для станции)
5	nginx:1.17	Веб сервер
6	clickhouse_http_api:master	API для доступа к данным измерений в ClickHouse
7	param-threshold:master	Сервис обеспечивает проверку выхода параметров за назначенные им пределы (так называемые уставки)
8	event-writer:master	Сервис записи событий по данным параметров со станции
9	equipment-type-go:master	Каталог оборудования
10	go-unit_nodes_api:master	Каталог узлов агрегатов. Каталог дефектов агрегатов. API автоматических правил
11	go-doc_storage_api:master	НСИ (нормативно-справочная информация) сервис каталога документов
12	data_delay_detector:master	Сервис обнаружения задержки поступления данных в систему
13	api-gw:master	Шлюз API сервисов go
14	web_php	API доступа для взаимодействия с клиентским приложением
15	system_system	Сервис с набором служб для работы системы
16	web_node	Фронт (Frontend)
17	redis:alpine	NOsql БД для кэширования данных
18	rabbitmq:3-management	Брокер сообщений, для обработки событий
19	centrifugo/centrifugo:v2.2.6	Сервис вебсокетов (frontend websockets).

2.4. Создание баз данных.

Перед набором команд создания БД в ClickHouse командой clickhouse-client запускаем интерактивную консоль для управления СУБД Clickhouse

2.4.1. Создание БД.

Запускаем команду CREATE DATABASE PranaRAW

2.4.2. Создание таблиц.

Далее создаются две таблицы (PranaRAW.ParamValues и PranaRAW.MsetValues)

1)

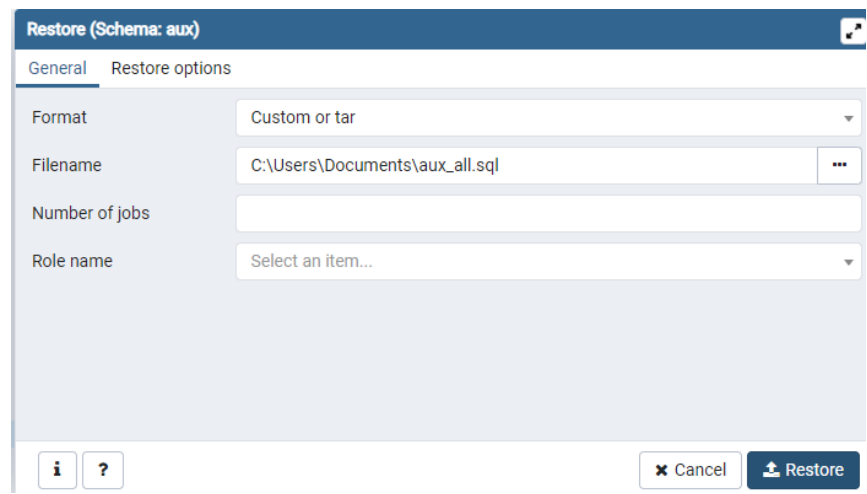
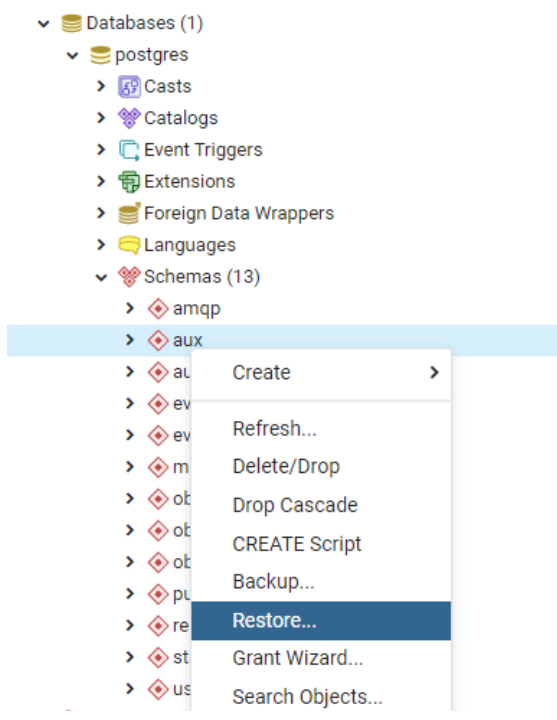
```
CREATE TABLE PranaRAW.ParamValues
```

Перв. Примен.	<pre>(`TS` UInt64 CODEC(Delta(8), ZSTD(1)), `Sid` UInt16, `Uid` UInt16, `Pid` UInt32, `Val` Float64, `Quality` Decimal(3, 0), `measurement_type` Decimal(3, 0) DEFAULT 1, `OpCTS` UInt64 CODEC(Delta(8), ZSTD(1)))</pre>					
Справ. №	<pre>ENGINE = MergeTree() PARTITION BY toMonth(toDate(TS)) ORDER BY (Sid, Uid, Pid, measurement_type, TS) SETTINGS index_granularity = 8192 2) CREATE TABLE PranaRAW.MsetValues (`TS` UInt64 CODEC(Delta(8), ZSTD(1)), `Sid` UInt16, `Uid` UInt16, `Pid` UInt32, `BasedOnUid` UInt16, `BasedOnMT` Decimal(3, 0) DEFAULT 1, `Val` Float64, `Quality` Decimal(3, 0), `measurement_type` Decimal(3, 0) DEFAULT 1, `TopDiscr` Array(Tuple(UInt16, UInt32, UInt16, Float64)))</pre>					
Подп. и дата	<pre>ENGINE = MergeTree() PARTITION BY toYYYYMM(toDateTime(TS / 1000)) ORDER BY (Sid, Uid, Pid, measurement_type, TS) SETTINGS index_granularity = 8192</pre>					
Инв. № дубл	<p>2.4.3. Создание материализованных представлений.</p> <p>Создаются два материализованных представления (PranaRAW.last_positions_mset и PranaRAW.last_positions)</p>					
Взам. инв. №						
Подп. и дата						
Инв. № подл						
Изм	Лист	№ докум.	Подп.	Дата	<p align="center">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>	Лист
						17

Перв. Примен.	<p>1) CREATE MATERIALIZED VIEW PranaRAW.last_positions_mset (`max_TS` UInt64 CODEC(Delta(8), ZSTD(1)), `Sid` UInt16, `Uid` UInt16, `Pid` UInt32, `BasedOnUid` UInt16, `BasedOnMT` Decimal(3,0) DEFAULT 1, `_Val` Float64, `_TopDiscr` Array(Tuple(UInt16, UInt32, UInt16, Float64)), `Quality` Decimal(3,0), `measurement_type` Decimal(3,0)) ENGINE = ReplacingMergeTree(max_TS) PARTITION BY tuple() ORDER BY (Sid, Uid, measurement_type, Pid, BasedOnUid, BasedOnMT, Quality) SETTINGS index_granularity = 8192 AS SELECT max(TS) AS max_TS, Sid, Uid, Pid, BasedOnUid, BasedOnMT, argMax(Val, TS) AS _Val, argMax(TopDiscr, TS) AS _TopDiscr, Quality, measurement_type FROM PranaRAW.MsetValue GROUP BY Sid,</p>					
Справ. №						
Подп. и дата						
Инв. № дубл						
Взам. инв. №						
Подп. и дата						
Инв. № подл						
Изм	Лист	№ докум.	Подп.	Дата	Инструкция по установке © АО «ПОТЕК Диджитал Солюшнс» Конфиденциально	Лист
						18

Перв. Примен.					<pre> measurement_type, Uid, Pid, BasedOnUid, BasedOnMT, Quality </pre>																		
Справ. №					<pre> 2) CREATE MATERIALIZED VIEW PranaRAW.last_positions (`max_TS` UInt64 CODEC(Delta(8), ZSTD(1)), `Sid` UInt16, `Uid` UInt16, `Pid` UInt32, `_Val` Float64, `Quality` Decimal(3,0), `measurement_type` Decimal(3,0)) ENGINE = ReplacingMergeTree(max_TS) PARTITION BY tuple() ORDER BY (Sid, Uid, measurement_type, Pid, Quality) SETTINGS index_granularity = 8192 AS SELECT max(TS) AS max_TS, Sid, Uid, Pid, argMax(Val, TS) AS _Val, Quality, measurement_type FROM PranaRAW.ParamValues GROUP BY Sid, measurement_type, Uid, Pid, Quality </pre>																		
Подп. и дата																							
Инв. № дубл																							
Взам. инв. №																							
Подп. и дата																							
Инв. № подл					<p>2.4.4. Перенос основных схем.</p>																		
Изм																							
Лист	Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально																					Лист	
№ докум.																						19	
Подп.																							
Дата																							

Перв. Примен.	<p>На сервере с БД PostgreSQL переносятся основные схемы из представленных архивов (demo-schema.rar).</p> <p>2.4.4.1. Перечень основных схем:</p> <p>1 aux,- содержит системные данные, данные расчетов</p> <p>2 events – содержит события на станциях</p> <p>3 mset_python – содержит математические расчеты</p> <p>4 obj_station_name – содержит статические данные о станции, имена агрегатов, параметров, их адреса в системе АСУ</p> <p>5 resources – содержит картинки, рисунки мнемосхем</p> <p>6 stations – содержит описание станций и сопоставление мнемосхем агрегатам</p> <p>7 users – пользователи системы</p>					
	Справ. №	<p>2.4.4.2. Создаем схемы, перечисленные выше:</p> <pre>CREATE SCHEMA aux AUTHORIZATION postgres; CREATE SCHEMA events AUTHORIZATION postgres; CREATE SCHEMA mset AUTHORIZATION postgres; CREATE SCHEMA obj_station_name AUTHORIZATION postgres; CREATE SCHEMA resources AUTHORIZATION postgres; CREATE SCHEMA stations AUTHORIZATION postgres; CREATE SCHEMA users AUTHORIZATION postgres;</pre>				
Подп. и дата		<p>2.4.4.3. Разворачивание бэкапов.</p> <p>Далее с помощью pgAdmin разворачиваем бэкапы системных папок в наши созданные схемы</p>				
	Инов. № дубл					
Взам. инв. №						
	Подп. и дата					
Инов. № подл						
	Изм	Лист	№ докум.	Подп.	Дата	<p align="center">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>
					20	



Эти действия производим с каждой нужной нам схемой.

2.5. Копирование и настройка переменных для файлов Docker Compose.

В комплекте дистрибутива предоставляются compose файлы, имеющие расширение *.yml
Т.к. эти файлы содержат параметры конфигурации, определяемые конкретной инсталляцией – следует эти параметры настроить, непосредственно отредактировав значения в файлах.

Ниже указаны также листинги вышеуказанных файлов с примечаниями.

2.5.1. Листинг файла docker-compose.yml

Для запуска веб составляющей сервиса, в нем конфигурируются пути к папкам монтируемым в контейнер и адресация сервисов (по необходимости, по умолчанию адреса закомментированы все кроме nginx).

```
version: '3'
```


Перв. Примен.	<pre> - ./custom_log/prana.api:/var/log/prana.api/custom_log/ #- Директория монтирования логов - ./resources:/var/www/resources #- Директория для загрузок мнемосхем и системных файлов - /mnt/data/PyService/share_folder/pymset:/var/www/resources/pymset #- Директория для обмена данных с математическими модулями - /mnt/data/export_click/files:/var/www/resources/export #- Директория файлов выгрузок из системы - /mnt/bin_write/bin_files:/var/www/resources/data #- Директория для бинарных файлов (в новой версии может быть устаревшей) - /mnt/data/PyService/share_folder/pymset/station_models:/var/www/resources/station_models #- Директория с моделями аналитики. working_dir: /usr/share/nginx/html networks: - проху node: build: ./prana.client restart: always volumes: - ./prana.client:/var/www #- Директория с кодом фронта networks: - проху centrifugo: image: centrifugo/centrifugo:v2.2.6 command: centrifugo -c config.json --port=8000 --log_level=debug --debug --address=0.0.0.0 ports: - "8001:8000" volumes: - ./centrifugo/config.json:/centrifugo/config.json #- файл конфигурации networks: - проху networks: проху: external: name: проху Для удобства сервисы объединяются одной подсетью: docker network create --driver=bridge --subnet=172.160.0.0/16 --IP-range=172.160.10.0/24 --gateway=172.160.10.254 проху 2.5.2. Листинг compose-api-gw-master.yml version: '2' networks: default: external: name: проху </pre>					
Справ. №						
Подп. и дата						
Инв. № дубл						
Взам. инв. №						
Подп. и дата						
Инв. № подл						
Изм	Лист	№ докум.	Подп.	Дата	<p style="text-align: center;">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>	<p style="text-align: center;">Лист 23</p>

Перв. Примен.	services:					
	<pre> api-gw: container_name: \${CONTAINER_NAME} #- указать имя контейнера image: \${IMAGE} #- указать образ restart: on-failure stop_signal: SIGINT stop_grace_period: 5s privileged: true environment: &env - CONTAINER_NAME=\${CONTAINER_NAME} #- указать имя контейнера - STAGE_NAME=\${STAGE_NAME} 2.5.3. Листинг compose-clickhouse_http_api-master.yml version: '3' services: clickhouse_http_api: container_name: clickhouse_http_api image: \${IMAGE} #- указать образ restart: always stop_signal: SIGINT stop_grace_period: 5s privileged: true networks: - проху environment: &env - MYAPP_KH_HOST=\${MYAPP_KH_HOST} #- IP адрес БД ClickHouse - MYAPP_KH_PORT=\${MYAPP_KH_PORT} #- порт БД ClickHouse - MYAPP_KH_PROTOCOL=\${MYAPP_KH_PROTOCOL} #- по умолчанию tcp - MYAPP_KH_SCHEME=\${MYAPP_KH_SCHEME} #- схема в БД ClickHouse - MYAPP_KH_USER=\${MYAPP_KH_USER} #- пользователь БД ClickHouse - MYAPP_KH_PASS=\${MYAPP_KH_PASS} #- пароль БД ClickHouse - MYAPP_PG_HOST=\${MYAPP_PG_HOST} #- IP адрес БД PostgreSQL - MYAPP_PG_PORT=\${MYAPP_PG_PORT} #- порт БД PostgreSQL - MYAPP_PG_SCHEME=\${MYAPP_PG_SCHEME} #- схема в БД PostgreSQL - MYAPP_PG_USER=\${MYAPP_PG_USER} #- пользователь БД PostgreSQL - MYAPP_PG_PASS=\${MYAPP_PG_PASS} #- пароль БД PostgreSQL - MYAPP_SERV_PORT=\${MYAPP_SERV_PORT} #- прослушиваемый порт приложения (3002) - PUBLIC_DEPLOY_KEY=\${PUBLIC_DEPLOY_KEY} #- необязательный параметр ports: - 0.0.0.0:7890:3002 volumes: - "/home/dev/WinShare/Bin9b/data:/app/binaries" #- директория с файлами бинарными </pre>					
Справ. №						
Подп. и дата						
Инв. № дубл						
Взам. инв. №						
Подп. и дата						
Инв. № подл						
Изм	Лист	№ докум.	Подп.	Дата	Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально	Лист
						24

Перв. Примен.		<pre> networks: proxy: external: name: proxy </pre>				
Справ. №		<p>2.5.4. Листинг compose-collector_cfg_distributor-master.yml</p> <pre> version: '2' networks: default: external: name: proxy services: config-distributor: container_name: \${CONTAINER_NAME} #- имя контейнера image: \${IMAGE} #- образ restart: on-failure stop_signal: SIGINT stop_grace_period: 5s privileged: true environment: &env - CONTAINER_NAME=\${CONTAINER_NAME} #- имя контейнера - STAGE_NAME=\${STAGE_NAME} #- имя стенда/окружения (prod, dev ...) - BROKER_ADDR=\${BROKER_ADDR} #- адрес Kafka - BROKER_PORT=\${BROKER_PORT} #- порт Kafka - OWN_API_PORT=\${OWN_API_PORT} #- порт API сервиса - OWN_API_ACCESS_KEY=\${OWN_API_ACCESS_KEY} #- ключ доступа к API - BROKER_AUTH_TYPE=\${BROKER_AUTH_TYPE} #- необязательный параметр - BROKER_AUTH_CREDS=\${BROKER_AUTH_CREDS} #- необязательный параметр - API_USER=\${API_USER} #- логин авторизации в системе прана - API_PASS=\${API_PASS} #- пароль авторизации в системе прана - API_BASE_URL=\${API_BASE_URL} #- адрес базовой API php - STATION_KEY_LIST=\${STATION_KEY_LIST} #- перечень ключей станций - SVC_ACCESS_TOKENS=\${SVC_ACCESS_TOKENS} #- сервисный ключ доступа - API_GATE_BASE_URL=\${API_GATE_BASE_URL} #- адрес сервиса api-gw (настраивается в nginx) </pre>				
Подп. и дата		<p>2.5.5. Листинг compose-data_delay_detector-master.yml</p> <pre> version: '2' services: data_delay_detector: container_name: \${CONTAINER_NAME} #- имя контейнера image: \${IMAGE} #- образ контейнера restart: on-failure stop_signal: SIGINT </pre>				
Инв. № дубл						
Взам. инв. №						
Подп. и дата						
Инв. № подл						
Изм	Лист	№ докум.	Подп.	Дата	<p align="center">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>	Лист
						25

Перв. Примен.	<pre> stop_grace_period: 5s privileged: true network_mode: "host" environment: &env - CONTAINER_NAME=\${CONTAINER_NAME} #- имя контейнера - STAGE_NAME=\${STAGE_NAME} #- необязательный параметр - INSTANCE_NAME=\${INSTANCE_NAME} #- по умолчанию main - APP_ENV=\${APP_ENV} #- по умолчанию prod - SRC_BROKERS=\${SRC_BROKERS} #- адрес (источник) Kafka вместе с портом - DST_BROKERS=\${DST_BROKERS} #- адрес (приемник) Kafka вместе с портом - EVENT_TOPIC=\${EVENT_TOPIC} #- наименование topic (data.events) - CFG_API_TKN=\${CFG_API_TKN} #- ключ авторизации - CFG_API_URL=\${CFG_API_URL} #- адрес API конфигуратора (настраивается в nginx) </pre>					
	Справ. №	<p>2.5.6. Листинг compose-equipment-type-go-master.yml</p> <pre> version: '2' networks: default: external: name: проху services: equipment_type: container_name: \${CONTAINER_NAME} #- имя контейнера image: \${IMAGE} - образ restart: on-failure stop_signal: SIGINT stop_grace_period: 5s privileged: true environment: &env - CONTAINER_NAME=\${CONTAINER_NAME} #- имя контейнера - STAGE_NAME=\${STAGE_NAME} #- необязательный параметр - DB1_HOST=\${DB1_HOST} #- IP адрес БД PostgreSQL - DB1_PORT=\${DB1_PORT} #- порт БД PostgreSQL - DB1_USER=\${DB1_USER} #- пользователь БД PostgreSQL - DB1_PASSWORD=\${DB1_PASSWORD} #- пароль БД PostgreSQL - DB1_DBNAME=\${DB1_DBNAME} #- имя БД PostgreSQL - DB2_HOST=\${DB2_HOST} #- IP адрес БД PostgreSQL - DB2_PORT=\${DB2_PORT} #- порт БД PostgreSQL - DB2_USER=\${DB2_USER} #- пользователь БД PostgreSQL - DB2_PASSWORD=\${DB2_PASSWORD} #- пароль БД PostgreSQL - DB2_DBNAME=\${DB2_DBNAME} #- имя БД PostgreSQL - PORT=\${PORT} #- порт сервиса(26111) </pre>				
Подп. и дата		Изм. № дубл	Взам. инв. №	Подп. и дата	Изм. № подл	<p align="center">Инструкция по установке © АО «РОТЕК Диджитал Солюинс» Конфиденциально</p>
					26	
Изм	Лист	№ докум.	Подп.	Дата		

Перв. Примен.	<ul style="list-style-type: none"> - ACCESS_SECRET=\${ACCESS_SECRET} #- секрет доступа - CENTRIFUGO_IP_STAND=\${CENTRIFUGO_IP_STAND} #- IP адрес centryfugo - CENTRIFUGO_PORT_STAND=\${CENTRIFUGO_PORT_STAND} #- порт centryfugo - CENTRIFUGO_API_KEY_STAND=\${CENTRIFUGO_API_KEY_STAND} #- ключ доступа centryfugo - CENTRIFUGO_SECRET_KEY_STAND=\${CENTRIFUGO_SECRET_KEY_STAND} #- секрет centryfugo - CENTRIFUGO_IP=\${CENTRIFUGO_IP} #- IP адрес centryfugo - CENTRIFUGO_PORT=\${CENTRIFUGO_PORT} #- порт centryfugo - CENTRIFUGO_API_KEY=\${CENTRIFUGO_API_KEY} #- ключ доступа centryfugo - CENTRIFUGO_SECRET_KEY=\${CENTRIFUGO_SECRET_KEY} #- секрет centryfugo - REDIS_CONFIG_SCHEME=\${REDIS_CONFIG_SCHEME} #- схема redis (tcp) - REDIS_CONFIG_HOST=\${REDIS_CONFIG_HOST} #- адрес redis - REDIS_CONFIG_PORT=\${REDIS_CONFIG_PORT} #- порт redis - GOCH_HOST=\${GOCH_HOST} #- адрес API go_ch (берем из – clickhouse_http_api-master) - KAFKA_HOST=\${KAFKA_HOST} #- адрес Kafka вместе с портом - KAFKA_TECH_EVENTS_TOPIC=\${KAFKA_TECH_EVENTS_TOPIC} #- название topic в Kafka 				Справ. №		
	<p>2.5.7. Листинг compose-event-writer-master.yml</p> <pre> version: '2' services: event-writer: container_name: \${CONTAINER_NAME} #- имя контейнера image: \${IMAGE} #- образ restart: on-failure stop_signal: SIGINT stop_grace_period: 5s privileged: true network_mode: "host" environment: &env - CONTAINER_NAME=\${CONTAINER_NAME} #- имя контейнера - STAGE_NAME=\${STAGE_NAME} #- необязательный параметр - APP_ENV=\${APP_ENV} #- по умолчанию production - DB_HOST=\${DB_HOST} #- IP адрес БД PostgreSQL - DB_PORT=\${DB_PORT} #- порт БД PostgreSQL - DB_USER=\${DB_USER} #- пользователь БД PostgreSQL - DB_PASS=\${DB_PASS} #- пароль БД PostgreSQL - DB_NAME=\${DB_NAME} #- имя БД PostgreSQL - DB_TABLE_NAME=\${DB_TABLE_NAME} #- имя таблицы (по умолчанию events.events) - BROKER_ADDR=\${BROKER_ADDR} #- адрес Kafka - EVENT_TOPICS=\${EVENT_TOPICS} #- название topic Kafka - RABBIT_HOST=\${RABBIT_HOST} #- IP адрес rabbit - RABBIT_PORT=\${RABBIT_PORT} #- порт rabbit - RABBIT_USER=\${RABBIT_USER} #- пользователь rabbit - RABBIT_PASS=\${RABBIT_PASS} #- пароль rabbit </pre>						
Подп. и дата					Инв. № дубл		
Подп. и дата					Взам. инв. №		
Инв. № подл					Подп. и дата		
Изм	Лист	№ докум.	Подп.	Дата	Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально		Лист
							27

Перв. Примен.					<p>- RABBIT_EVENT_ROUTING=\${RABBIT_EVENT_ROUTING} #-по умолчанию p_events_v3</p> <p>- RABBIT_EVENT_EXCHANGE=\${RABBIT_EVENT_EXCHANGE}} #-по умолчанию postgres_exchange</p> <p>networks:</p> <p> proxy:</p> <p> external:</p> <p> name: proxy</p> <p>2.5.8. Листинг compose-go-analytics-master.yml</p> <p>version: '2'</p> <p>networks:</p> <p> default:</p> <p> external:</p> <p> name: proxy</p> <p>services:</p> <p> go-analytics:</p> <p> container_name: \${CONTAINER_NAME} #- имя контейнера</p> <p> image: \${IMAGE} - образ</p> <p> restart: on-failure</p> <p> stop_signal: SIGINT</p> <p> stop_grace_period: 25s</p> <p> privileged: true</p> <p> volumes:</p> <p> - /home/dev/WinShare/Stations/./app/mdl/ #- директория монтирования моделей</p> <p> environment: &env</p> <p> - CONTAINER_NAME=\${CONTAINER_NAME} #- имя контейнера</p> <p> - STAGE_NAME=\${STAGE_NAME} #- необязательный параметр</p> <p> - WATCH_UNITS=\${WATCH_UNITS} #- необязательный параметр</p> <p> - REDIS_HOST=\${REDIS_HOST} #- адрес сервиса redis</p> <p> - REDIS_PORT=\${REDIS_PORT} #- порт сервиса redis</p> <p> - PRANA_API_BASE_URL=\${PRANA_API_BASE_URL} #- адрес API (http://ip_domain/api/core)</p> <p> - PRANA_API_USERNAME=\${PRANA_API_USERNAME} #- пользователь системы прана</p> <p> - PRANA_API_PASS=\${PRANA_API_PASS} #- пароль системы прана</p> <p> - CFG_API_TOKEN=\${CFG_API_TOKEN} #- токен авторизации cfg</p> <p> - CFG_API_URL=\${CFG_API_URL} #- API конфигуратора (http://ip_domain /cfgman)</p> <p> - BROKER_ADDR=\${BROKER_ADDR} #- адрес Kafka</p> <p> - SRC_TOPICS=\${SRC_TOPICS} #- topic источника (mset.slices)</p> <p> - MDL_BASE_PATH=\${MDL_BASE_PATH} #- директория с моделями</p> <p> - POINT_TOPIC_OUT=\${POINT_TOPIC_OUT} #- topic аналитики (mset.points)</p> <p> - META_POINT_TOPIC_OUT=\${META_POINT_TOPIC_OUT} #- topic аналитики (mset.metapoints)</p> <p> - EVENT_TOPIC_OUT=\${EVENT_TOPIC_OUT} #- topic аналитики (data.events)</p> <p> - TOP_DISCREPANCY_TOPIC_OUT=\${TOP_DISCREPANCY_TOPIC_OUT} #- topic аналитики (mset.topdiscr)</p> <p> - DATA_STATE_TOPIC_OUT=\${DATA_STATE_TOPIC_OUT} #- topic аналитики (statemsg)</p> <p> - TECH_EVENT_TOPIC=\${TECH_EVENT_TOPIC} #- topic аналитики (ipc.tech.events)</p>
Справ. №					
Подп. и дата					
Инв. № дубл					
Взам. инв. №					
Подп. и дата					
Инв. № подл					
Изм	Лист	№ докум.	Подп.	Дата	<p style="text-align: center;">Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально</p>
					Лист 28

Перв. Примен.	<pre> - API_GATE_URL=\${API_GATE_URL} #- API api_gw (http://ip_domain /gate) - INSTANCE_ID=\${INSTANCE_ID} #- по умолчанию (production) - APP_ENV=\${APP_ENV} #- по умолчанию (production) - API_LISTEN_PORT=\${API_LISTEN_PORT} #- порт сервиса (22233) 2.5.9. Листинг compose-go-doc_storage_api-master.yml version: '3' services: go-doc_storage_api: container_name: go-doc_storage_api image: \${IMAGE} – образ restart: always stop_signal: SIGINT stop_grace_period: 5s privileged: true networks: - проху environment: &env - MYAPP_PG_HOST=\${MYAPP_PG_HOST} #- IP адрес БД PostgreSQL - MYAPP_PG_PORT=\${MYAPP_PG_PORT} #- порт БД PostgreSQL - MYAPP_PG_DBNAME=\${MYAPP_PG_DBNAME} #- имя БД PostgreSQL - MYAPP_PG_USER=\${MYAPP_PG_USER} #- пользователь БД PostgreSQL - MYAPP_PG_PASS=\${MYAPP_PG_PASS} #- пароль БД PostgreSQL - MYAPP_SERV_PORT=\${MYAPP_SERV_PORT} #- порт сервиса (3003) - PUBLIC_DEPLOY_KEY=\${PUBLIC_DEPLOY_KEY} #- необязательный параметр networks: проху: external: name: проху </pre>					
Справ. №						
			Име. № дубл			
Подп. и дата			Взам. име. №			
			Подп. и дата			
			Име. № подл			
			Изм			
			Лист			
			№ докум.			
			Подп.			
			Дата			
			Инструкция по установке © АО «РОТЕК Диджитал Солюшнс» Конфиденциально			
						Лист
						29

Перв. Примен.		<ul style="list-style-type: none"> - MYAPP_PG_HOST=\${MYAPP_PG_HOST} #- IP адрес БД PostgreSQL - MYAPP_PG_PORT=\${MYAPP_PG_PORT} #- порт БД PostgreSQL - MYAPP_PG_DBNAME=\${MYAPP_PG_DBNAME} #- имя БД PostgreSQL - MYAPP_PG_USER=\${MYAPP_PG_USER} #- пользователь БД PostgreSQL - MYAPP_PG_PASS=\${MYAPP_PG_PASS} #- пароль БД PostgreSQL - MYAPP_KHGOAPI_HOST=\${MYAPP_KHGOAPI_HOST} #- IP адрес хоста с API - MYAPP_KHGOAPI_PORT=\${MYAPP_KHGOAPI_PORT} #- порт API кх (7890) - MYAPP_SERV_PORT=\${MYAPP_SERV_PORT} #- порт сервиса (3004) - PUBLIC_DEPLOY_KEY=\${PUBLIC_DEPLOY_KEY} #- необязательный параметр
Справ. №		<p>networks:</p> <p>проху:</p> <p>external:</p> <p>name: proxy</p> <p>2.5.11. Листинг compose-kafka_slicer-master.yml</p> <pre>version: '2' services: kafka_slicer: container_name: \${CONTAINER_NAME} #- имя контейнера image: \${IMAGE} #- образ restart: on-failure stop_signal: SIGINT stop_grace_period: 5s privileged: true network_mode: "host" environment: &env - CONTAINER_NAME=\${CONTAINER_NAME} #- имя контейнера - STAGE_NAME=\${STAGE_NAME} #- необязательный параметр - BROKER_ADDR=\${BROKER_ADDR} #- адрес Kafka - SRC_TOPIC_LIST=\${SRC_TOPIC_LIST} #- topic Kafka (filtered.points) - DST_TOPIC=\${DST_TOPIC} #- topic Kafka (mset.slices) - SLICE_INTERVAL_SEC=\${SLICE_INTERVAL_SEC} #- интервал сообщений topic (10) - INSTANCE_NAME=\${INSTANCE_NAME} #- имя инстанса (production_main) - HANDLE_SID_LIST=\${HANDLE_SID_LIST} #- необязательный параметр - PRANA_API_URL=\${PRANA_API_URL} #- адрес API php - PRANA_API_LOGIN=\${PRANA_API_LOGIN} #- логин пользователя ПРАНА - PRANA_API_PASS=\${PRANA_API_PASS} #- пароль пользователя ПРАНА</pre>
Подп. и дата		
Инь. № дубл		
Взам. инв. №		
Подп. и дата		
Инь. № подл		

Перв. Примен.		<pre> restart: on-failure stop_signal: SIGINT stop_grace_period: 5s privileged: true network_mode: "host" environment: &env - CONTAINER_NAME=\${CONTAINER_NAME} #- имя контейнера - STAGE_NAME=\${STAGE_NAME} #- необязательный параметр - BROKER_ADDR=\${BROKER_ADDR} #- адрес Kafka - TOPIC_IN=\${TOPIC_IN} #- topic входящий (filtered.points) - DB_HOST=\${DB_HOST} #- адрес базы clickhouse - DB_PORT=\${DB_PORT} #- порт базы clickhouse - DB_SCHEME=\${DB_SCHEME} #- схема базы clickhouse (tcp) - DB_USER=\${DB_USER} #- пользователь - DB_PASS=\${DB_PASS} #- пароль - DB_NAME=\${DB_NAME} #- имя БД (PranaRAW) - BATCH_SIZE=1200 - DB_FLUSH_DELAY_MS=1500 - APP_ENV=\${APP_ENV} #- по умолчанию (production) - TOPICS_IN=\${TOPICS_IN} #- topic Kafka (filtered.points,mset.metapoints,mset.points,mset.topdiscr) - CONSUMER_GROUP_NAME= kh_writer networks: proxy: external: name: proxy </pre>				
Справ. №		<pre> 2.5.13. Листинг compose-param-threshold-master.yml </pre>				
		<pre> version: '2' services: param-threshold: container_name: \${CONTAINER_NAME} #- имя контейнера image: \${IMAGE} #- образ restart: on-failure stop_signal: SIGINT stop_grace_period: 5s privileged: true network_mode: "host" environment: &env - CONTAINER_NAME=\${CONTAINER_NAME} #- имя контейнера - STAGE_NAME=\${STAGE_NAME} #- необязательный параметр - WATCH_SID_LIST=\${WATCH_SID_LIST} #- необязательный параметр - BROKER_ADDR_LIST=\${BROKER_ADDR_LIST} #- адрес Kafka - CFG_TOPIC_NAME=\${CFG_TOPIC_NAME} #- topic конфигурации (необязательный параметр) </pre>				
Подп. и дата						
Инв. № дубл						
Взам. инв. №						
Подп. и дата						
Инв. № подл		<p style="text-align: center;">Инструкция по установке © АО «РОТЕК Диджитал Солюинс» Конфиденциально</p>				
Изм	Лист	№ докум.	Подп.	Дата		Лист 31

- APP_ENV=\${APP_ENV} #- по умолчанию (production)
- POINT_TOPIC_LIST=\${POINT_TOPIC_LIST} #- topic Kafka (filtered.points)
- EVENT_TOPIC=\${EVENT_TOPIC} #- topic Kafka (data.events)
- POINT_BROKER_ADDR_LIST=\${POINT_BROKER_ADDR_LIST} #- адрес Kafka
- CFG_API_URL=\${CFG_API_URL} #- API конфигуратора (http://ip_domain /cfgman)
- CFG_API_TOKEN=\${CFG_API_TOKEN} #- токен ав

2.6. Команды запуска.

После подготовки баз данных, файлов проекта, файлов моделей и файлов конфигураций приступаем к запуску.

Каждый из них нужно запустить под командой 'docker-compose up -f имя файла'.
Запуск нужно сделать в указанной последовательности.

№№	Название файла
1	compose-collector_cfg_distributor-master.yml
2	docker-compose.yml
3	compose-api-gw-master.yml
4	compose-clickhouse_http_api-master.yml
5	compose-data_delay_detector-master.yml
6	compose-equipment-type-go-master.yml
7	compose-event-writer-master.yml
8	compose-go-analytics-master.yml
9	compose-go-doc_storage_api-master.yml
10	compose-go-unit_nodes_api-master.yml
11	compose-kafka_slicer-master.yml
12	compose-kh_writer-master.yml
13	compose-param-threshold-master.yml

Первым запускаем веб часть, из директории /web в которой находится файл docker-compose.yml запускаем -> **docker-compose up -d** (для того чтобы потушить контейнеры, если вдруг неправильно настроена конфигурация **docker-compose down**)

Далее переходим в директорию **go_cont** и поочередно запускаем контейнеры **docker-compose -f NAME_FILE.YML up -d** (пример **docker-compose -f compose-api-gw-master.yml up -d**)

После того, как все сервисы успешно запущены пользователь может открыть ПО в браузере по адресу **http://IP_адрес_сервера_приложений**. Должен открыться интерфейс ПО «Система Прогностики и Удаленного Мониторинга (СПиУМ) «ПРАНА» с предложением ввести учетные данные.

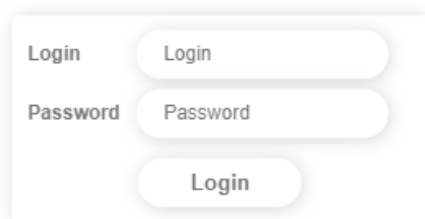


Рисунок 1. Скриншот окна с предложением ввести учетные данные

После введения логина (expert) и пароля (expert_pass), пользователь попадает в раздел «Обзор» См. рисунок ниже

Перев. Примен.

Справ. №

Подп. и дата

Инв. № дубл

Взам. инв. №

Подп. и дата

Инв. № подл

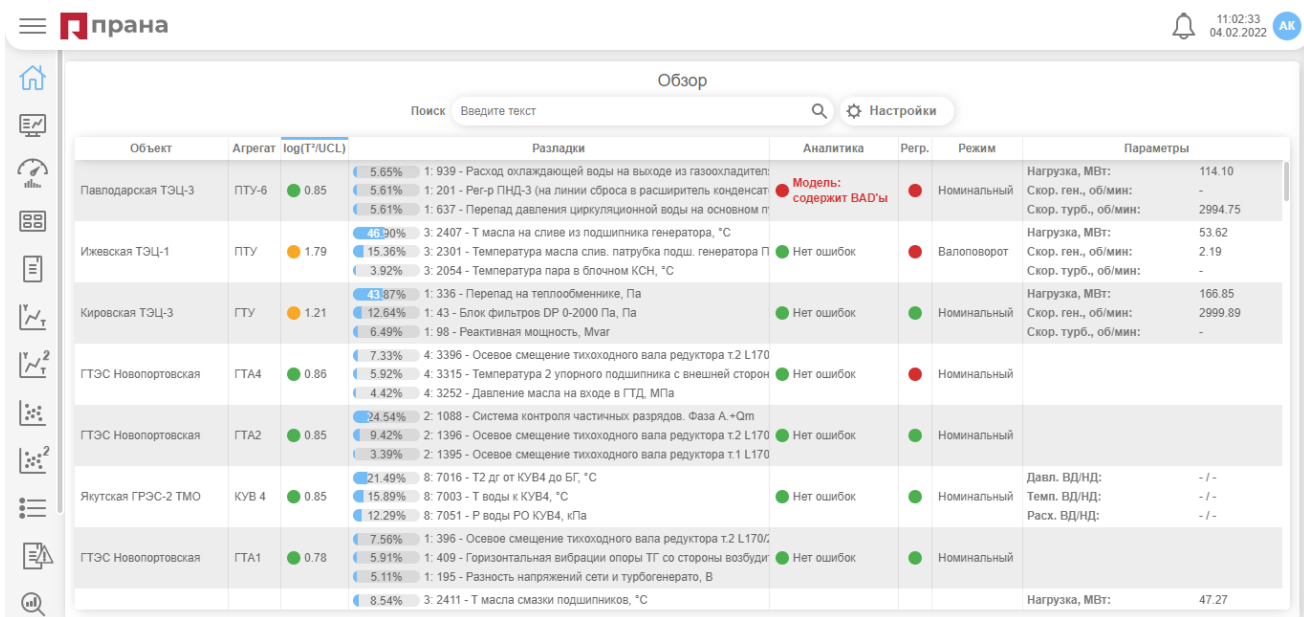


Рисунок 2. Интерфейс после открытия ПО

Далее в работе с ПО «Система Прогностики и Удаленного Мониторинга (СПИУМ) «ПРАНА»» следует руководствоваться документом

Краткое руководство по работе с «Системой Прогностики и Удаленного Мониторинга (СПИУМ) «ПРАНА»», размещенного по ссылке:

<https://prana-system.com/sistema-prognostiki-i-udalennogo-monitoringa-spium-prana/>

Правообладатель готов продемонстрировать работоспособность Программного обеспечения «Система Прогностики и Удаленного Мониторинга (СПИУМ) «ПРАНА»».

Для этого просим связаться по телефону или электронной почте

Клочков Александр Павлович

a.klochkov@rotec.digital

Тел.: +7(495)-644-3460 доб.2358

Моб.: +7(903)-712-5293

Изм	Лист	№ докум.	Подп.	Дата
-----	------	----------	-------	------