ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ПЛАТФОРМА «ТЕЛЕМАТИКА.ПРО»

ЭКСПЛУАТАЦИОННАЯ ДОКУМЕНТАЦИЯ

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Москва 2025

Аннотация

В настоящем документе приводится Руководство пользователя к Программному обеспечению платформы «Телематика. Про».

Содержание

1 Введение 4

1.1 Область применения 4

1.2 Краткое описание возможностей 4

1.3 Уровень подготовки пользователя 5

1.4 Перечень эксплуатационной документации, с которыми необходимо ознакомиться пользователю 5

2 Назначение и условия применения 6

2.1 Виды деятельности, функции, для автоматизации которых предназначено данное средство автоматизации 6

2.2 Условия, при соблюдении которых обеспечивается применение средства автоматизации в соответствии с назначением 7

2.2.1 T-Store — Базы данных и хранение данных 7

2.2.2 T-Flow — Потоковая обработка данных 8

2.2.3 T-GeoCore — Геопространственные сервисы 8

2.2.4 T-Control — Оркестрация и управление 9

2.2.5 T-Pipeline — Хранение файлов и CI/CD 10

2.2.6 T-Support — Коммуникации и поддержка 10

2.2.7 T-SmartMon — Мониторинг и аналитика 10

3 Подготовка к работе 12

3.1 Порядок загрузки данных и программ 12

3.2 Порядок проверки работоспособности 12

3.2.1 T-Store — Базы данных и хранение данных 12

3.2.2 T-Flow — Потоки и обработка 14

3.2.3 T-GeoCore — Геопространственные сервисы 14

3.2.4 T-Control — Оркестрация и управление 15

3.2.5 T-Pipeline — Хранение файлов и CI/CD 15

3.2.6 T-Support — Коммуникации и поддержка 15

3.2.7 T-SmartMon — Мониторинг 16

4 Аварийные ситуации 17

1 Введение

1.1 Область применения

Платформа «Телематика.Про» предназначена для решения следующих задач:

1. Обработка и хранение пространственных данных.

Платформа «Телематика.Про» предназначена для работы с геопространственными данными с использованием компонентов PostGIS, Valhalla, mbtileserver, и Graphhopper. Данные инструменты позволяют эффективно хранить, обрабатывать и визуализировать географическую информацию, что критически важно для систем навигации, логистики и управления транспортом.

2. Обработка потоковых данных и событий.

Платформа обрабатывает большие объемы потоковых данных в реальном времени. Это позволяет создавать системы мониторинга, аналитики и реагирования на события, такие как изменения в данных, поступающих от IoT-устройств или других источников.

3. Масштабируемость и оркестрация.

Использование Kubernetes, Docker и containerd обеспечивает Платформе высокую степень масштабируемости и гибкости в управлении контейнеризированными приложениями. Это позволяет легко развертывать, масштабировать и управлять микросервисной архитектурой для распределенных систем.

4. Мониторинг и аналитика.

Компоненты (Zabbix, Prometheus, Loki, Grafana, Kibana и Victoria) предоставляют инструменты для мониторинга, сбора метрик, логирования и визуализации данных. Это позволяет разработчикам и операторам Платформы быстро выявлять и устранять проблемы, а также проводить глубокий анализ производительности системы.

5. Хранение и управление данными.

Платформа поддерживает различные типы данных, включая реляционные (PostgreSQL), документоориентированные (MongoDB), поисковые (Elasticsearch) и аналитические (Clickhouse). Это позволяет гибко выбирать подходящее хранилище для конкретных задач, обеспечивая высокую производительность и надежность.

1.2 Краткое описание возможностей

Платформа «Телематика.Про» обеспечивает:

- Гибкость и масштабируемость. Компоненты Платформы позволяют легко адаптироваться к изменяющимся требованиям и масштабировать системы в зависимости от нагрузки.

- Высокая производительность. Использование специализированных инструментов для аналитики или для маршрутизации обеспечивает высокую скорость обработки данных.
- Надежность и отказоустойчивость. Компоненты обеспечивают высокую доступность и устойчивость к сбоям.
- Универсальность. Платформа поддерживает различные типы данных и сценарии использования, что делает ее применимой для широкого круга задач.
- Интеграция и экосистема. Все компоненты интегрируются между собой, что упрощает разработку и поддержку сложных систем.

1.3 Уровень подготовки пользователя

Для работы пользователи должны обладать следующими навыками и квалификацией:

а) навыки работы с компьютерами и периферийными устройствами;
б) умение пользоваться средствами операционной системы и оперировать ею через стандартные интерфейсы;
в) навыки работы с командной строкой;
г) Понимание репликации, шардинга, резервного копирования;
д) знание основ работы реляционных и нереляционных СУБД.

1.4 Перечень эксплуатационной документации, с которыми необходимо ознакомиться пользователю

Перед началом работы пользователям необходимо ознакомиться с документом по установке и эксплуатации ПО, а также с настоящим Руководством пользователя.

2 Назначение и условия применения

2.1 Виды деятельности, функции, для автоматизации которых предназначено данное средство автоматизации

- обработка геопространственных данных, таких как маршруты, точки интереса и геозоны;
- автоматизация задач, таких как резервное копирование и очистка данных;
- управление большими таблицами через партиционирование;
- построение маршрутов и анализ графа;
- мониторинг производительности запросов;
- оптимизация таблиц для повышения производительности;
- хранение документоориентированных данных, таких как профили пользователей или настройки;
- поиск и анализ текстовых данных, например, логов или событий;
- аналитика больших объемов данных в реальном времени;
- оркестрация потоков данных между микросервисами;
- автоматизация потоков данных, включая их обработку и маршрутизацию;
- захват изменений в базах данных (CDC) для обработки в реальном времени;
- построение маршрутов и расчет времени прибытия;
- хранение и предоставление векторных и растровых карт;
- генерация векторных карт для использования в приложениях;
- оптимизация маршрутов для транспорта и логистики;
- сбор метрик и мониторинг состояния системы;
- централизованное хранение и анализ логов;
- визуализация данных и создание дашбордов;
- отслеживание ошибок и исключений в приложениях;
- управление контейнерами и микросервисами;
- обнаружение сервисов и управление конфигурациями;
- координация распределенных систем;
- хранение больших объемов данных, таких как файлы или резервные копии;
- автоматизация процессов CI/CD;
- управление обращениями пользователей;
- управление электронной почтой;
- интеграция с другими сервисами.

2.2 Условия, при соблюдении которых обеспечивается применение средства автоматизации в соответствии с назначением

Все требования приведены для одного экземпляра сервиса. При расчёте ресурсов на кластер необходимо учитывать масштабирование и резервирование.

Для кластерных и отказоустойчивых развёртываний рекомендуется закладывать ресурсы с запасом.

При необходимости производительность отдельных сервисов может быть масштабирована горизонтально или вертикально.

Поддерживаются развертывания как на физических, так и на виртуальных серверах, а также в облачной инфраструктуре.

2.2.1 T-Store — Базы данных и хранение данных

PostgreSQL / PostGIS

CPU: 8 ядер и более.

RAM: 32 ГБ и более.

Диск для ОС: 100 ГБ SSD.

Диск для данных: 500 ГБ SSD (желательно RAID1).

Сеть: 1 x 10 GbE

MongoDB

CPU: 8 ядер и более.

RAM: 32 ГБ и более.

Диск для ОС: 100 ГБ SSD.

Диск для данных: 500 ГБ SSD (RAID1, желательно отдельные диски для journal и данных).

Сеть: 1 x 10 GbE.

Elasticsearch

CPU: 8 ядер и более.

RAM: 32 ГБ и более.

Диск для ОС: 100 ГБ SSD.

Диск для данных: 500 ГБ SSD (NVMe рекомендуется).

Сеть: 1 x 10 GbE.

ClickHouse

CPU: 8 ядер и более.

RAM: 32 ГБ и более.

Диск для ОС: 100 ГБ SSD.

Диск для данных: 1 ТБ SSD/NVMe (скоростные).

Сеть: 1 x 10 GbE.

2.2.2 T-Flow — Потоковая обработка данных

Kafka

CPU: 8 ядер и более.

RAM: 32 ГБ и более.

Диск для ОС: 100 ГБ SSD.

Диск для данных: 500 ГБ SSD (RAID1, рекомендуется быстрые диски).

Сеть: 2 x 10 GbE.

Дополнительно: для HA — нечетное количество брокеров (3+).

NiFi

CPU: 8 ядер и более.

RAM: 32 ГБ и более.

Диск для ОС: 100 ГБ SSD.

Диск для данных: 500 ГБ SSD.

Сеть: 1 x 10 GbE.

Debezium

CPU: 4 ядра и более.

RAM: 8 ГБ и более.

Диск для ОС и данных: 100 ГБ SSD.

Сеть: 1 x 1 GbE.

2.2.3 T-GeoCore — Геопространственные сервисы

Valhalla

CPU: 4 ядра и более.

RAM: 16 ГБ и более.

Диск для ОС и данных: 100 ГБ SSD (с учётом объёма картографических данных).

Сеть: 1 x 1 GbE.

Mbtileserver

CPU: 2 ядра и более.

RAM: 8 ГБ и более.

Диск для ОС и данных: 50 ГБ SSD (с учётом объёма карт).

Сеть: 1 x 1 GbE.

Graphhopper

CPU: 4 ядра и более.

RAM: 16 ГБ и более.

Диск для ОС и данных: 100 ГБ SSD.

Сеть: 1 x 1 GbE.

2.2.4 T-Control — Оркестрация и управление

Kubernetes (master/worker)

CPU: 8 ядер и более.

RAM: 32 ГБ и более.

Диск для ОС: 100 ГБ SSD.

Диск для контейнеров: 500 ГБ SSD.

Сеть: 2 x 10 GbE.

Docker / Containerd

CPU: 4 ядра и более.

RAM: 8 ГБ и более.

Диск: 100 ГБ SSD.

Сеть: 1 x 1 GbE.

Consul

CPU: 4 ядра и более.

RAM: 8 ГБ и более.

Диск: 50 ГБ SSD.

Сеть: 1 x 1 GbE.

Zookeeper

CPU: 4 ядра и более.

RAM: 8 ГБ и более.

Диск: 50 ГБ SSD.

Сеть: 1 x 1 GbE.

Дополнительно: для отказоустойчивости — нечетное число инстансов (3+).

Harbor

CPU: 8 ядер и более.

RAM: 16 ГБ и более.

Диск для ОС: 100 ГБ SSD.

Диск для хранения образов: 500 ГБ SSD.

Сеть: 1 x 10 GbE.

2.2.5 T-Pipeline — Хранение файлов и CI/CD

Minio

CPU: 4 ядра и более.

RAM: 16 ГБ и более.

Диск для ОС: 100 ГБ SSD.

Диск для хранилища: 500 ГБ SSD/NVMe.

Сеть: 1 x 10 GbE.

Gitlab-runner

CPU: 4 ядра и более.

RAM: 8 ГБ и более.

Диск: 100 ГБ SSD.

Сеть: 1 x 1 GbE.

2.2.6 T-Support — Коммуникации и поддержка

FreeScout

CPU: 2 ядра и более.

RAM: 4 ГБ и более.

Диск: 50 ГБ SSD.

Сеть: 1 x 1 GbE.

Mailcow

CPU: 4 ядра и более.

RAM: 8 ГБ и более.

Диск для ОС: 100 ГБ SSD.

Диск для почты: 200 ГБ SSD (с учетом объема ящиков).

Сеть: 1 x 1 GbE.

2.2.7 T-SmartMon — Мониторинг и аналитика

Zabbix, Prometheus, VictoriaMetrics, Alertmanager

CPU: 4 ядра и более.

RAM: 8 ГБ и более.

Диск: 100 ГБ SSD.

Сеть: 1 x 1 GbE.

Loki

CPU: 4 ядра и более.

RAM: 8 ГБ и более.

Диск: 100 ГБ SSD.

Сеть: 1 x 1 GbE.

Grafana, Kibana

CPU: 2 ядра и более.

RAM: 4 ГБ и более.

Диск: 50 ГБ SSD.

Сеть: 1 x 1 GbE.

Sentry, Glitchtip

CPU: 4 ядра и более

RAM: 8 ГБ и более

Диск: 50 ГБ SSD

Сеть: 1 x 1 GbE

3 Подготовка к работе

3.1 Порядок загрузки данных и программ

Доступ к установленному ПО осуществляется посредством SSH соединения к одной из необходимых ВМ:

- 10.20.31.11 pgmain-m01
- 10.20.31.12 pgbouncer
- 10.20.31.13 kafka
- 10.20.31.14 nifi
- 10.20.31.15 consul
- 10.20.31.16 mongo
- 10.20.31.17 els
- 10.20.31.18 click
- 10.20.31.19 zabbix
- 10.20.31.20 sentry
- 10.20.31.21 loki
- 10.20.31.22 kube-m01
- 10.20.31.23 kube-w01
- 10.20.31.25 minio
- 10.20.31.26 gitlab-runner
- 10.20.31.27 mailcow
- 10.20.31.28 harbor

3.2 Порядок проверки работоспособности

Базовые проверки работоспособности сервисов и приложений производятся на виртуальных машинах, где они запущенны. Для этого необходимо подключиться к виртуальной машине по ssh и произвести проверки, описанные ниже.

3.2.1 T-Store — Базы данных и хранение данных

PostgreSQL + PostGIS

CLI:

```
psql -U postgres -d yourdb -c "SELECT ST_AsText(ST_GeomFromText('POINT(30 10)'));"
```

Python:

```
import psycopg2
conn = psycopg2.connect("dbname=yourdb user=postgres password=yourpass")
cur = conn.cursor()
cur.execute("SELECT ST_AsText(ST_GeomFromText('POINT(30 10)'))")
```

```
print(cur.fetchone())
```

MongoDB

CLI:
```
mongosh --eval "db.adminCommand({ping: 1})"
```

В случае успеха, в терминал вернётся ответ:

```
{ ok: 1 }
```

Python:
```
from pymongo import MongoClient
client = MongoClient("mongodb://user:pass@localhost:27017/")
db = client.testdb
db.test.insert_one({"check": True})
assert db.test.find_one({"check": True})
```

Elasticsearch

curl:
```
curl -X GET http://localhost:9200/_cluster/health?pretty -u "user:password"
```

В случае успеха, в терминал вернётся ответ:

```
{
  "cluster_name" : "elasticsearch",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 1,
  "number_of_data_nodes" : 1,
  "active_primary_shards" : 6,
  "active_shards" : 6,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 0,
  "unassigned_primary_shards" : 0,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 100.0
}
```

Python:
```
from elasticsearch import Elasticsearch
es = Elasticsearch("http://localhost:9200")
assert es.ping()
res = es.search(index="*", query={"match_all": {}})
print(res)
```

ClickHouse

CLI:
```
clickhouse-client --query="SELECT 1"
```

В случае успеха, в терминал вернётся ответ:

```
1
```

Python:

```
from clickhouse_driver import Client
client = Client('localhost')
print(client.execute('SELECT 1'))
```

3.2.2 T-Flow — Потоки и обработка

Kafka (консольные тесты)

После подключения по ssh на виртуальную машину с kafka необходимо перейти в директорию */opt/kafka/bin* и далее выполнить команды для проверок ниже:

Создание топика:
```
kafka-topics.sh --create --topic test --bootstrap-server localhost:9092
```

Публикация:
```
echo "hello" | kafka-console-producer.sh --broker-list localhost:9092 --topic test
```

Чтение:
```
kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning --timeout-ms 2000
```

Zookeeper

```
echo "ruok" | nc localhost 2181
# Ожидаем ответ "imok"
```

NiFi

```
curl -s http://localhost:8080/nifi | grep -i "NiFi"
```

Debezium (проверка топика)

```
kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic dbserver1.inventory.customers --from-beginning
```

3.2.3 T-GeoCore — Геопространственные сервисы

Valhalla

```
curl -s -X POST http://localhost:30188/route --header 'Content-Type: application/json' \
--data '{
    "locations": [
        {
            "lat": 55.99225011541637,
            "lon": 37.21668858879568
        },
        {
            "lat": 56.860169268438966,
            "lon": 35.92686827935145
        }
    ],
    "costing": "auto",
    "directions_options": {
        "units": "km",
        "language": "ru-RU"
    }
```

```
}' | grep route
```

Mbtileserver

```
curl http://localhost:30180/services/russia/tiles/13/4952/2562.pbf --output tile.pbf
file tile.pbf
```

OpenMapTiles

Проверка тайлов с помощью картографического клиента (Leaflet или MapLibre) или:

```
curl -I http://localhost:8080/services/russia/tiles/13/4952/2562.pbf
```

Graphhopper

```
curl
"http://localhost:8989/route?point=52.54,13.4050&point=52.54,13.4060&vehicle=car&loca
le=en"
```

3.2.4 T-Control — Оркестрация и управление

Kubernetes

```
kubectl get nodes
kubectl get pods -A
```

Python (kubernetes API):

```
from kubernetes import client, config
config.load_kube_config()
v1 = client.CoreV1Api()
print([node.metadata.name for node in v1.list_node().items])
```

Docker / Containerd

```
docker run hello-world
nerdctl run hello-world
```

Consul

```
curl http://localhost:8500/v1/catalog/services
```

3.2.5 T-Pipeline — Хранение файлов и CI/CD

MinIO

```
mc alias set local http://localhost:9000 minioadmin minioadmin
mc mb local/testbucket
mc ls local/testbucket
```

GitLab Runner

```
gitlab-runner verify
```

3.2.6 T-Support — Коммуникации и поддержка

FreeScout (вручную через UI)

- Открыть веб-интерфейс;
- Создать тикет;
- Получить уведомление на e-mail.

Mailcow

```
docker exec -it mailcowdockerized_postfix-mailcow_1 mailq
```

Проверка через Web UI (RainLoop / SOGo).

3.2.7 T-SmartMon — Мониторинг

Prometheus

```
curl http://localhost:9090/-/healthy
```

VictoriaMetrics

```
curl http://localhost:8428/metrics
```

Zabbix

Проверить в UI, что агент виден и онлайн.

Loki + Promtail

```
curl -G -s "http://localhost:3100/loki/api/v1/query" \
  --data-urlencode 'query={job="varlogs"}'
```

Grafana

Приложение развернуто в кластере kubernetes.

```
curl -I http://localhost:30190
```

Kibana

Приложение развернуто в кластере kubernetes

```
curl http://localhost:5601/api/status | jq '.status.overall'
```

Sentry / Glitchtip

- Через UI: создать проект;
- Отправить тестовую ошибку из приложения;
- Убедиться, что она появилась в интерфейсе.

4 Аварийные ситуации

При возникновении ошибок в работе ПО пользователю следует обратиться в службу технической поддержки ПО. В обращении следует указать описание аварийной ситуации, действия, приведшие к ее возникновению, по возможности приложить скриншоты и тексты выдаваемых сообщений.