

Руководство по установке и настройке Системы мониторинга мостов в реальном времени

1. Общие сведения

Система представляет собой набор микросервисов, разворачиваемых в Docker на виртуальных машинах или выделенных серверах (VPS/VDS). Для работы микросервисов требуется предварительное развертывание дополнительных сторонних сервисов. Сторонние сервисы являются бесплатными и не имеют ограничений по типам (коммерческое/частное) или регионам использования.

Всего можно выделить 3 группы сервисов:

- 1) Сторонние бесплатные – требуют первоочередной установки и настройки
- 2) Вендорские (ZetLab, Mango) – должны быть предоставлены/развернуты поставщиком
- 3) Собственные микросервисы, собственно представляющие собой Систему

1.1. Сторонние сервисы, требующиеся для работы Системы

- 1.1.1. Consul – сервис обнаружения и мониторинга статуса прочих сервисов; хранилище настроек
- 1.1.2. PostgreSQL – база данных
- 1.1.3. Kafka – журнал сообщений для обмена данными между сервисами
- 1.1.4. SMTP-сервер – почтовый сервер для отправки уведомлений по email
- 1.1.5. Nginx – веб-сервер и реверс-прокси
- 1.1.6. Docker – система контейнеризации сервисов и приложений
- 1.1.7. OS Linux – операционная система, предпочтительно (но не обязательно) семейства RHEL: CentOS Stream или российский Гослинукс (<https://goslinux.fssp.gov.ru/>). Также возможно использование операционной системы семейства Debian: Ubuntu или российской Astra Linux (<https://astralinux.ru/>)

1.2. Вендорские сервисы, наличие и доступность которых обеспечиваются компанией-поставщиком

- 1.2.1. SCADA-софт – специальная программа для преобразования сырых данных с датчиков в пригодные для использования
- 1.2.2. БД «zetlab.com» – база данных с данными, полученными от SCADA-софта
- 1.2.3. Манго АПИ – программный интерфейс, позволяющий отправлять SMS и звонки

1.3. Собственные микросервисы

- 1.3.1. Sensor Data Extractor – извлечение данных из БД «zetlab.com» и отправки их в Kafka
- 1.3.2. User Service – хранилище пользователей и работа с ними
- 1.3.3. Gateway Auth – шлюз входящих запросов, выполняющий авторизацию и маршрутизацию
- 1.3.4. Alert Generator – прослушивание сообщений из журнала Kafka, проверка на необходимость генерации событий, генерация событий и нотификаций, их сохранение и отправка в Kafka для сервисов доставки оповещений
- 1.3.5. Notification Config Service – конфигурация оповещений для событий (каналы доставки, адресаты)
- 1.3.6. Notification Delivery Call – доставка оповещений звонком
- 1.3.7. Notification Delivery Email – доставка оповещений по e-mail
- 1.3.8. Notification Delivery SMS – доставка оповещений по SMS
- 1.3.9. Notification Delivery Telegram – доставка оповещений через Telegram-бота
- 1.3.10. Historical Service – хранение и работа с историческими данными
- 1.3.11. Frontend – предоставление веб-интерфейса
- 1.3.12. File to S3 Data Copier – перекладывает файлы с FTP Zetlab в S3-хранилище системы для обучения нейросети предсказывать приближение к пороговым значениям

2. Порядок установки

2.1. Предусловия

Перед началом развертывания убедитесь, что у вас имеются следующие виртуальные или выделенные серверы.

№	Назначение	Процессор	ОЗУ	ПЗУ
1	Gateway, Consul, Frontend, UI	4 ядра 3,3 ГГц	4 Гб	40 Гб
2	Kafka	8 ядер 3,3 ГГц	16 Гб	200 Гб
3	PostgreSQL	8 ядер 3,3 ГГц	8 Гб	200 Гб
4	Alerts module	8 ядер 3,3 ГГц	4 Гб	40 Гб
5	Services	4 ядра 3,3 ГГц	8 Гб	40 Гб

На серверах необходимо выполнить стандартную настройку доступов, безопасности и др. При установке сервисов необходимо обеспечивать открытие требуемых портов с использованием стандартных инструментов (iptables, firewalld, ufw...). Установить Docker.

ЗАМЕЧАНИЕ. Сервер №1 является точкой входа и должен быть доступен извне (или из целевой сети), тогда как остальные серверы могут находиться в приватной сети и быть недоступны извне (но должны быть доступны с сервера №1).

ЗАМЕЧАНИЕ 2. Consul нежелательно делать публичным, и рекомендуется разворачивать его на сервере №5, а также рекомендуется публиковать фронтенд системы по https протоколу, но данное руководство разрабатывалось для полностью защищенного развертывания системы, когда никакой сервер не виден снаружи, поэтому для удобства мониторинга все ui-инструменты развернуты на сервере №1, а сам сервер №1 находится в закрытой сети (VPN).

2.1. Как вносить настройки в Consul KV

- 1) Открыть http://адрес_сервера_1:8500/ui/dc1/kv
- 2) Нажать Create
- 3) В поле Key or folder указать config/server_name/data
- 4) В выпадающем списке выбрать YAML
- 5) Вставить указанный yaml-конфиг из руководства
- 6) Изменить значения требуемым образом при необходимости
- 7) Нажать Save

< Key / Values

New Key / Value

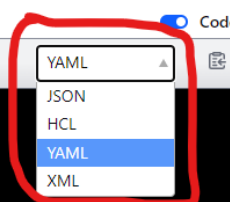
Key or folder

config/user-service/data

To create a folder, end a key with /

Value

1



8)

Save

Cancel

2.2. Развертывание сервисов

2.2.1. Consul (сервер №1)

```
mkdir /data
mkdir /data/consul
cd /data/consul

#положить файл docker-compose-consul.yml и переименовать в docker-compose.yml

docker compose up -d
```

Для проверки открыть в браузере http://адрес_сервера_1:8500/ui

2.2.2. Интерфейс Kafka (сервер №1)

```
mkdir /data/kafka-ui
cd /data/kafka-ui

#положить файл docker-compose-kafka-ui.yml и переименовать в docker-compose.yml
#открыть его для редактирования и в строках 8-9 изменить айпи-адрес
# на соответствующий серверу №2

docker compose up -d
```

Для проверки открыть в браузере http://адрес_сервера_1:8080

2.2.3. PostgreSQL (сервер №3)

```
mkdir /data
mkdir /data/postgresql

cd /data/postgresql

#положить файл docker-compose-postgresql.yml и переименовать в docker-compose.yml
#при необходимости открыть файл и изменить юзернейм и/или пароль на требуемые
#также можно изменить способ задания юзернейма и пароля на secrets

docker compose up -d
```

Убедиться, что порт 5432 открыт и принимает входящие подключения

2.2.4. Kafka (сервер №2)

```
mkdir /data
mkdir /data/kafka
cd /data/kafka

#положить файл docker-compose-kafka.yaml и переименовать в docker-compose.yaml
#открыть его для редактирования и в строке 25 изменить айпи-адрес
# на соответствующий серверу №2

docker compose up -d
```

Для проверки открыть в браузере http://адрес_сервера_1:8080 и убедиться, что кластер появился в UI

2.2.5. User Service (сервер №5)

```
#здесь и далее переменная $DOCKER_REGISTRY означает ваш приватный докер-репозиторий,  
# куда сложены собранные образы сервисов. Перед выполнением дальнейших команд  
# необходимо выполнить docker login в ваш репозиторий $DOCKER_REGISTRY.  
  
mkdir /data  
mkdir /data/user-service  
cd /data/user-service  
  
#положить файл docker-compose-user-service.yaml и переименовать в docker-compose.yaml  
#открыть его для редактирования и указать айпи-адреса и юзернейм/пароль от PostgreSQL.  
  
docker compose up -d
```

Для проверки открыть в браузере http://адрес_сервера_1:8500/ui/dc1/services и убедиться, что user-service зеленый.

ЗАМЕЧАНИЕ. Все последующие сервисы кроме фронтенда проверять аналогичным образом.

2.2.6. Gateway Auth (сервер №1)

```
mkdir /data/gateway-auth  
cd /data/gateway-auth  
  
#положить файл docker-compose-gateway-auth.yaml и переименовать в docker-compose.yaml  
#открыть его для редактирования и указать айпи-адреса и юзернейм/пароль от PostgreSQL.  
  
docker compose up -d
```

2.2.7. Sensor Data Extractor (сервер №5)

```
mkdir /data/sensor-data-extractor  
cd /data/sensor-data-extractor  
  
#положить файл docker-compose-sensor-data-extractor.yaml и переименовать в docker-  
compose.yaml  
#открыть его для редактирования и указать айпи-адреса и данные для доступа к БД «Zetlab»  
  
docker compose up -d
```

2.2.8. Alert Generator (сервер №4)

Перед запуском требуется внести конфигурацию в Consul Key-Value!

```
notification:  
  objectName: Рогачёвский мост
```

Запуск:

```
mkdir /data  
mkdir /data/alert-generator  
cd /data/alert-generator  
  
#положить файл docker-compose-alert-generator.yaml и переименовать в docker-compose.yaml  
#открыть его для редактирования и указать айпи-адреса и данные для доступа к PostgreSQL  
  
docker compose up -d
```

2.2.9. Notification Config Service (сервер №5)

```
mkdir /data/notification-config-service
cd /data/notification-config-service

#положить файл docker-compose-notification-config-service.yaml и переименовать в docker-
compose.yaml
#открыть его для редактирования и указать айпи-адреса и данные для доступа к PostgreSQL

docker compose up -d
```

2.2.10. Historical Service (сервер №5)

```
mkdir /data/historical-service
cd /data/historical-service

#положить файл docker-compose-historical-service.yaml и переименовать в docker-
compose.yaml
#открыть его для редактирования и указать айпи-адреса и данные для доступа к PostgreSQL

docker compose up -d
```

2.2.11. Notification Delivery Sms/Email/Call/Telegram (сервер №4)

Перед запуском требуется внести конфигурацию для каждого сервиса в Consul Key-Value!

notification-delivery-call:

```
call-config:
  # инфа из личного кабинета, вкладка https://lk.mango-
office.ru/400233969/400325774/api-vpbx/connector
  apiKey: "ramc7lkchhy9475sr3dt956vbzs3fucn"
  # инфа из личного кабинета, вкладка https://lk.mango-
office.ru/400233969/400325774/api-vpbx/connector
  apiSalt: "u5hl806whc8xnkpn3grii3zfvz73vcxnf"
  # инфа из личного кабинета, вкладка https://lk.mango-
office.ru/400233969/400325774/api-vpbx/connector
  host: "https://app.mango-office.ru/vpbx/"
  # ИД созданного в настройках кампании параметра, значением которого будет текст для
озвучивания
  parameter-with-text: "Комментарий"
  required-campaign-status: 2

template:
  text: |
    {objectName}: {message}! Расположение: {sensorLocation}. Рекомендуется
    {recommendation}.
```

notification-delivery-sms:

```
sms-config:
  # 11 - это идентификатор сотрудника из ЛК манго офиса
  senderId: "11"
  # это дефолтное, настроить тут https://lk.mango-
office.ru/400233969/400325774/additional-settings/sms
  senderName: "COM-PRO"
  # инфа из личного кабинета, вкладка https://lk.mango-
office.ru/400233969/400325774/api-vpbx/connector
  apiKey: "ramc7lkchhy9475sr3dt956vbzs3fucn"
  # инфа из личного кабинета, вкладка https://lk.mango-
office.ru/400233969/400325774/api-vpbx/connector
  apiSalt: "u5hl806whc8xnkpn3grii3zfvz73vcxnf"
  # инфа из личного кабинета, вкладка https://lk.mango-
office.ru/400233969/400325774/api-vpbx/connector
  host: "https://app.mango-office.ru/vpbx/"
template:
  text: |
    Событие {severity}! Рекомендация: {recommendation}
```

notification-delivery-email:

```
email-config:
  subject: "Оповещение системы безопасности" # тайтл письма
  sender: "bridge@maxima.ru" # отправитель письма, должен быть верифицирован на SMTP
шлюзе
  host: smtp.maxima.ru
  port: 587
  username: bridge@maxima.ru
  password: ^QooC0Ea~D=D
  smtpAuth: true
  starttlsEnabled: true
  starttlsRequired: true
template:
  text: |
    Критичность: {severity}
    Объект: {objectName}
    Время: {moment}
    Датчик: {signal}
    Тип датчика: {sensorType}
    Расположение датчика: {sensorLocation}
    Сообщение: {message}
    Рекомендация: {recommendation}
```

notification-delivery-telegram:

```
template:
  text: |
    <b>Критичность:</b> {severity}
    <b>Объект:</b> {objectName}
    <b>Время:</b> {moment}
    <b>Датчик:</b> {signal}
    <b>Тип датчика:</b> {sensorType}
    <b>Расположение датчика:</b> {sensorLocation}
    <b>Сообщение:</b> {message}
    <b>Рекомендация:</b> {recommendation}
```

Запуск:

```
# обобщенная инструкция для четырех очень похожих сервисов
mkdir /data/*
cd /data/*

#положить файл docker-compose-notification-delivery-*.yaml и переименовать в docker-
compose.yaml
#открыть его для редактирования и указать айпи-адреса

docker compose up -d
```

2.2.12. Frontend (сервер №1)

```
mkdir /data/most-front
cd /data/most-front

#положить файл docker-compose-most-front.yaml и переименовать в docker-compose.yaml
#открыть его для редактирования и указать айпи-адрес Consul, если требуется (зависит от
версии)

docker compose up -d
```

Для проверки открыть в браузере http://адрес_сервера_1 (в Консале он отображаться не должен)

2.2.13. File to S3 Data Copier (сервер №5)

Перед запуском требуется внести конфигурацию в Consul Key-Value!

```
ftp:
  host: ftp.example.com
  port: 21
  username: yourFtpUsername
  password: yourFtpPassword
  path: /path/to/files

s3:
  bucket:
    name: your-s3-bucket-name
```

Запуск:

```
mkdir /data/file-to-s3-data-copier
cd /data/file-to-s3-data-copier

#положить файл docker-compose-file-to-s3-data-copier.yaml и переименовать в docker-
compose.yaml
#открыть его для редактирования и указать айпи-адреса

docker compose up -d
```