Документация на ПО

Оглавление

[1. Общее описание 3](#_Toc142041146)

[1.1 Обозначение и наименование программы 3](#_Toc142041147)

[1.2 Используемые языки программирования 3](#_Toc142041148)

[2. Функциональные характеристики 4](#_Toc142041149)

[2.1 Краткое описание функционала отдельных модулей 5](#_Toc142041150)

[3. Документация для эксплуатации программно-аппаратного комплекса 6](#_Toc142041151)

[3.1 Информация о требованиях к системе и окружению для работы программно-аппаратного комплекса 6](#_Toc142041152)

[3.2 Информация по выбору прокси-провайдера 7](#_Toc142041153)

[4. Инструкции по установке и настройке программно-аппаратного комплекса 8](#_Toc142041154)

[4.1 Установка warp10 8](#_Toc142041155)

[4.2 Генерация административного токена. 9](#_Toc142041156)

[4.3 Установка web-приложения для отображение пользовательского интерфейса. 11](#_Toc142041157)

[4.4 Установка сервиса мониторинга показателей. 14](#_Toc142041158)

# Общее описание

# 1.1 Обозначение и наименование программы

Наименование программы - “ Объективный Downdetector ”

# 1.2 Используемые языки программирования

* Python 3
* HTML/CSS
* JavaScript

# Функциональные характеристики

Цифровая платформа для обнаружения сбоев web-сервисов предназначена для автоматизации процесса обнаружения и мониторинга сбоев в web-сервисах. Она позволяет оперативно определять проблемы в работе сервисов и принимать меры по их устранению.

Платформа выполняет следующие функции:

1. **Сбор показателей работы исследуемого сервиса** - Платформа собирает метрики сервиса посредством HTTP-запросов режиме реального времени и позволяет накапливать необходимый для анализа объем данных
2. **Определение факта сбоя web-сервиса** - На основе собранных данных оперативно обнаруживает факт сбоя сервиса
3. **Определение загруженности web-сервиса** На основе собранных данных оперативно обнаруживает факты увеличения нагрузки на сервис путем анализа показателя latency
4. **Определение сетевых проблем** - На основе собранных данных о потерях в опорной сети оперативно обнаруживает факты о сбоях в сетевой инфраструктуре интернет-провайдеров
5. **Анализ данных** - Позволяет анализировать исторические данные на основе накопленных показателей и фактов сбоев, данная информация может быть использована для расследования инцидентов, улучшения качества сервиса

Цифровая платформа для обнаружения сбоев web-сервисов содержит следующие модули:

* Web-приложение для отображение пользовательского интерфейса
* Система хранения данных
* Сервис мониторинга показателей
* Модуль сопряжения (API)

# 2.1 Краткое описание функционала отдельных модулей

1. WEB-интерфейс с возможностью отображения информации в виде графиков и агрегированных данных.
2. База данных, представляющая собой экземпляр масштабируемой платформы Warp10, представляющую собой открытую базу данных хранения и обработки временных рядов.
3. Сервис сбора показателей реализует многопоточный, масштабируемый сбор данных по списку исследуемых web-сервисов и с использованием разных провайдеров связи.
4. Модуль сопряжения обрабатывает запросы из внешних источников посредством REST API.

# 3. Документация для эксплуатации программно-аппаратного комплекса

# Информация о требованиях к системе и окружению для работы программно-аппаратного комплекса

Система может быть установлена на любой сервер с **2 vCPU** и **4 GB RAM.** В качестве OC рекомендуется Debian 11.

Программные модули могут быть разнесены на разные сервера.

Наилучшим вариантом для организации работы программно-аппаратного комплекса является размещение его модулей на отдельных серверах. Это позволяет более эффективно управлять ресурсами каждого модуля и обеспечивает большую стабильность работы всего комплекса в целом.

Рекомендуется размещать модули "Web-приложение" и "Модуль сопряжения (API)" на одном сервере, так как они обеспечивают работу пользовательского интерфейса и взаимодействие с другими системами. Они используют общие ресурсы, поэтому их объединение на одном сервере уменьшит нагрузку на сеть и повысит производительность.

Модуль "Сервис мониторинга показателей" следует разместить на отдельном сервере, чтобы уменьшить нагрузку на сервера с другими модулями и обеспечить точную и своевременную передачу данных.

Модуль "Система хранения данных" отвечает за хранение и управление всей информацией, которую использует комплекс. Для обеспечения максимальной стабильности и безопасности работы этого модуля его также следует разместить на отдельном сервере.

Таким образом, разделение модулей программно-аппаратного комплекса на отдельные сервера позволяет оптимизировать работу каждого модуля и обеспечивает более эффективное использование ресурсов. Рекомендуется придерживаться данного подхода при установке и настройке комплекса.

# Информация по выбору прокси-провайдера

Платформа может быть использована как внутри уже имеющейся опорной сети, так и иметь единую точку выхода в сеть и использовать прокси-провайдеров для мониторинга сервиса с разных точек мира

В случае использования прокси-провайдера необходимо получить данные авторизации для каждого направления мониторинга и занести ее в соответствующие конфигурационные файлы. После этого платформа будет вести мониторинг через все имеющиеся точки.

# 4. Инструкции по установке и настройке программно-аппаратного комплекса

Для простоты описания установка будет производиться на один сервер.

# Установка warp10

Существует несколько вариантов установки окружения телеметрии - Standalone, docker и Distributed

Для целей простоты воспользуемся наиболее быстрым способом – установкой docker-версии.

В системе должен быть установлен docker последней версии.

Установим WARP10

docker run -d -p 8085:8080 -p 8081:8081 -e WARP10\_HEAP=8g -e WARP10\_HEAP\_MAX=8g --volume=/var/warp10:/data warp10io/warp10:latest

Параметры WARP10\_HEAP и WARP10\_HEAP\_MAX необходимо выставить в зависимости от конфигурации системы.

# Генерация административного токена.

Создадим конфигурационный файл токена:

mkdir -p /var/warp10/tokens

touch /var/warp10/tokens/admin.config.mc2

Вставим в файл admin.config.mc2 параметры генерации:

{

  'READ'

  {

    'id' 'tokenR'

    'type' 'READ'

    'application' $applicationName

    'owner'  $owner

    'issuance' NOW  1 ms /

    'expiry' $expirydate

    'labels' {}

    'attributes' {}

    'owners' [ $owner ]

    'producers' [ ]

    'applications' [ $applicationName ]

  } TOKENGEN

  'WRITE'

  {

    'id' 'tokenW'

    'type' 'WRITE'

    'application' $applicationName

    'owner'  $owner

    'producer' $owner

    'issuance' NOW   1 ms /

    'expiry' $expirydate

    'labels' {}

    'attributes' {}

  } TOKENGEN

}

Получаем id запущенного контейнера:

docker ps | grep 'warp10' | awk '{ print $1 }'

Войдем в запущенный контейнер:

docker exec -it [container-id] bash

Запустим процесс генерации (не забудьте заменить [your\_installed\_version] на uid версии jar-файла, установленного в docker):

java -cp /opt/warp10/bin/warp10-2.[your\_installed\_version].jar io.warp10.worf.TokenGen /opt/warp10/etc/conf.d/\* /data/tokens/admin.config.mc2 /data/tokens/admin.json

В выходном файле admin.json получим наш глобальный токен чтения и записи.

Токен чтения будет использоваться в конфигурационном файле Web-приложения для отображение пользовательского интерфейса. Токен записи будет использован в сервисе мониторинга показателей.

# Установка web-приложения для отображение пользовательского интерфейса.

В системе должен быть установлен python3 и пакет python3-pip.

Создадим путь до файлов ( /var/www/web ) и скопируем файлы из приложенного архива (web.tar.gz):

mkdir -p /var/www/web

tar xfm web.tar.gz -C /var/www/web

cd /var/www/web

Установим зависимости:

pip3 install -r back/requirements.txt

Добавим в конфигурационный файл, сгенерированный на шаге 3.1.1 токен чтения warp10:

[log]

file=downdetector.log

level=debug

[warp10]

url=http://127.0.0.1:8085/api/v0/exec

token=enter warp token # токен чтения warp10

Запуск рекомендуется осуществлять в связке supervisor + uvicorn:

pip3 install gunicorn

Произведем установку supervisor:

sudo apt install supervisor

sudo systemctl enable supervisor

Создадим конфигурационный файл supervisor:

sudo cat << EOF >> /etc/supervisor/conf.d/web.conf

[program:fastapi]

command=/usr/local/bin/gunicorn --log-file gun.log --log-level debug --timeout 90 -b 0.0.0.0:9000 -w 1 -k uvicorn.workers.UvicornWorker app:app

directory=/var/www/web/back/

user=root

autostart=true

autorestart=true

stopasgroup=true

killasgroup=true

EOF

Перезапустим supervisor

sudo supervisorctl reread

sudo supervisorctl reload

Произведем настройку nginx

В качестве обратного прокси-сервера для приложения webapp рекомендуется использовать последнюю стабильную версию Nginx.

<https://www.nginx.com/resources/wiki/start/topics/tutorials/install/>

sudo apt install nginx

Для генерации сертификатов можно воспользоваться letsencrypt. Замените <your\_fqdn> на используемое в системе доменное имя:

sudo apt install certbot python3-certbot-nginx

sudo certbot --nginx -d <your\_fqdn>

После установки, необходимо включить в действующую конфигурацию конфиг следующего содержания, заменив your\_server\_fqdn на используемое в системе доменное имя:

sudo cat << EOF >> /etc/nginx/sites-available/web

 upstream \_back {

    ip\_hash;

    server localhost:9000 weight=5;

 }

 server {

    listen 80;

    server\_name your\_server\_fqdn;

    error\_log /var/log/nginx/your\_server\_fqdn.log warn;

    access\_log /var/log/nginx/your\_server\_fqdn.access.log;

    return 301 https://your\_server\_fqdn\$request\_uri;

 }

 server {

    server\_name your\_server\_fqdn;

    root         /var/www/web;

    client\_max\_body\_size 100M;

    proxy\_connect\_timeout       600;

    proxy\_send\_timeout          600;

    proxy\_read\_timeout          600;

    send\_timeout                600;

    location ~ /back/  {

        proxy\_pass http://\_back;

        proxy\_set\_header Host \$host;

        proxy\_set\_header X-Forwarded-Proto \$scheme;

        proxy\_set\_header X-Real-IP \$remote\_addr;

        proxy\_set\_header X-Forwarded-For \$proxy\_add\_x\_forwarded\_for;

    }

     location / {

       root /var/www/web/front;

       index index.html;

       add\_header          Last-Modified \$date\_gmt;

       add\_header          Cache-Control 'no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0';

       if\_modified\_since   off;

       expires             off;

       etag                off;

       proxy\_cache off;

    }

    ssl\_certificate path\_to\_cert/fullchain.pem;

    ssl\_certificate\_key path\_to\_cert/privkey.pem;

    error\_log /var/log/nginx/your\_server\_fqdn.error.log warn;

    access\_log /var/log/nginx/your\_server\_fqdn.access.log;

    listen 443 ssl;

 }

EOF

Создадим символическую ссылку:

sudo ln -s /etc/nginx/sites-available/web /etc/nginx/sites-enabled/web

После изменений необходимо перечитать конфигурационные файлы nginx:

sudo nginx -s reload

После успешного запуска сервис начнет свою работу. Пользователь может подключиться через любой современный браузер и начать работу в системе. Пользовательский интерфейс будет доступен по адресу (https://your\_server\_fqdn)

# Установка сервиса мониторинга показателей.

В системе должен быть установлен python3 и пакет python3-pip.

Создадим путь до файлов (/opt/crawler) и скопируем файлы из приложенного архива (crawler.tar.gz) :

mkdir -p /opt/crawler

tar xfm crawler.tar.gz -C /opt/crawler

cd /opt/crawler

Установим зависимости:

pip3 install -r requirements.txt

Отредактируем конфигурационный файл main.conf:

[main]

threads=3 # кол-во потоков сбора

polling=60 # интервал опроса ресурсов

dump=300 # интервал сброса данных в БД

limit=15 # таймаут в секундах при осуществлении опроса

[log]

file=crawler.log # файл с логированием

level=debug # уровень логирования

[place]

provider=Ростелеком # название провайдера

place=Геленджик # местоположение

longitude=0 # координаты местоположения

lattitude=0 # координаты местоположения

[proxy] # список прокси-провайдеров

infantica=enabled # включение опроса через провайдера infantica (пример)

brightdata=disabled # отключение опроса через провайдера brightdata (пример)

[warp10]

url=http://127.0.0.1:8085/api/v0/update # адрес системы хранения warp10

token=<enter warp token here> # токен записи warp10 (см пункт 3.1.1)

Если предполагается использование прокси-провайдеров, то необходимо отредактировать файл proxy.conf, например:

[Infantica CentralTelegraf] # название прокси-провайдера и интернет-провайдера

proxy\_url=<add proxy url here> # адрес прокси (уточняется у прокси-провайдера)

provider=CentralTelegraf # название интернет-провайдера

place=Москва # местоположение

longitude=0 # координаты местоположения

lattitude=0 # координаты местоположения

status=enabled # статус (enabled, disabled)

[Infantica Rostelecom] # название прокси-провайдера и интернет-провайдера

proxy\_url=<add proxy url here> # адрес прокси (уточняется у прокси-провайдера)

provider=Rostelecom # название интернет-провайдера

place=Москва # местоположение

longitude=0 # координаты местоположения

lattitude=0 # координаты местоположения

status=enabled # статус (enabled, disabled)

При необходимости отредактируйте файл targets.conf для уточнения списка контролируемых сервисов

[yandex.ru] # название сервиса

Главная=https://yandex.ru # адрес мониторинга

[mail.ru]

Главная=https://yandex.ru

[sberbank.ru]

Главная=https://sberbank.ru

[Экосистема Сбера] # название сервиса

icon=sber # название миниатюры сервиса

salute.sber.ru=https://salute.sber.ru # адрес мониторинга

sberdevices.ru=https://sberdevices.ru # адрес мониторинга

jazz.sber.ru=https://jazz.sber.ru # адрес мониторинга

developers.sber.ru=https://developers.sber.ru # адрес мониторинга

Запустим приложение:

python3 crawler.py

Запуск рекомендуется осуществлять через supervisor:

Произведем установку supervisor (если не установлен ранее):

sudo apt install supervisor

sudo systemctl enable supervisor

Создадим конфигурационный файл supervisor:

sudo cat << EOF >> /etc/supervisor/conf.d/web.conf

[program:crawler]

command=python3 crawler.py

directory=/opt/crawler

user=root

autostart=true

autorestart=true

stopasgroup=true

killasgroup=true

EOF

Перезапустим supervisor

sudo supervisorctl reread

sudo supervisorctl reload

После успешного запуска сервис начнет свою работу и будет добавлять данные мониторинга в платформу warp10.