Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

Образовательная программа Прикладная математика и информатика

бакалавриат

Образовательная программа 09.03.04 «Программная инженерия»

**О Т Ч Е Т**

по технологической практике

в департаменте анализа данных и искусственного интеллекта, НИУ ВШЭ

Выполнил студент гр. БПИ196

Сахаров Н.Д.

*(ФИО)*

*(подпись)*

Руководитель практики от Организации

Старший преподаватель департамента анализа данных и искусственного интеллекта, м.н.с. МЛ ИССА, НИУ ВШЭ, Паринов Андрей Андреевич,

|  |  |  |
| --- | --- | --- |
| *21.07.2021* | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| *дата* | *оценка по 10 бальной шкале* | *подпись* |

Руководитель практики от департамента программной инженерии

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| *21.07.2021* | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| *дата* | *оценка по 10 бальной шкале* | *подпись* |

**Москва, 2021**

Содержание

[Цели и задачи практики 3](#_Toc77822657)

[Календарный план-график 3](#_Toc77822658)

[Обзор технологий 3](#_Toc77822659)

[Observability platform 3](#_Toc77822660)

[Примеры 5](#_Toc77822661)

[Обзор инструментов реализации 8](#_Toc77822662)

[Результаты 10](#_Toc77822663)

[Заключение 10](#_Toc77822664)

[Информационные источники 10](#_Toc77822665)

[Приложение А. 11](#_Toc77822666)

Аннотация

Отчет по технологической практике - документ, в котором описываются результаты, полученные в ходе выполнения задания по практике.

# Цели и задачи практики

В рамках практики необходимо изучить основы Python, изучить существующие подходы к мониторингу вычислительных систем и реализовать клиентский модуль системы мониторинга. Так как ведется работа в команде предполагается использование системы контроля версий (в данном случае Github). Все общение с командой и руководителем проводится в MS Teams.

Место прохождения практики

Место проведения практики: Национальный исследовательский университет «Высшая школа экономики», департамент анализа данных и искусственного интеллекта. Практика проходит в online формате с помощью teams.

# Календарный план-график

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Сроки проведения | Выполненные работы | Отметка руководителя о выполнении (подпись) |
| 1 | 01.07.2021 | Инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка. |  |
| 2 | 5.07.2021 | Обзор систем мониторинга и библиотек (алгоритмов). Разобраться с текущими наработками. |  |
| 3 | 12.07.2021 | Реализация первого прототипа клиентского модуля. |  |
| 4 | 15.07.2021 | Подготовка отчета. |  |
| 5 | 20.07.2021 | Доработка прототипа. Внесение правок в отчет. |  |

# Обзор технологий

## Observability platform

Постараемся понять, что же такое “observability platform”. В отношении IT технологий концепция обозримости была применена не так давно. Мы хотим наблюдать за различными системами. Изначально термин зародился в сфере систем управления. Мы  
можем считать, что умеем наблюдать за системой, если по внешним данным, которые она предоставляет, можем понять, что в ней происходит в выбранный момент за конечное время.

Достаточно абстрактно, посмотрим ближе на конкретно наш случай. Вычислительные системы, как правило, имеют журналы событий, логи, данные об ошибках и угрозах и т.д. Проанализировав все эти данные, мы можем понять, в каком состоянии находилась наша ВС в конкретный момент времени.

Но агрегация и анализ этих данных трудоемкий процесс, который хотелось бы ускорить. Кроме того, таких событий в каждый момент времени достаточной много, но, так же хочется узнавать о важных событиях как можно раньше. И так мы приходим к тому, что “observability platform” — это система мониторинга и анализа событий вычислительных систем. Она облегчает агрегацию данных и организует ее оперативный анализ.  
  
Можно расширить нашу вычислительную систему до совокупности разбросанных вычислительных узлов и сервисов. Существует множество решений некоторые из них более специфичны, некоторые предоставляют полный стек возможностей аналитики и  
мониторинга.

## Примеры

Graphana[12]

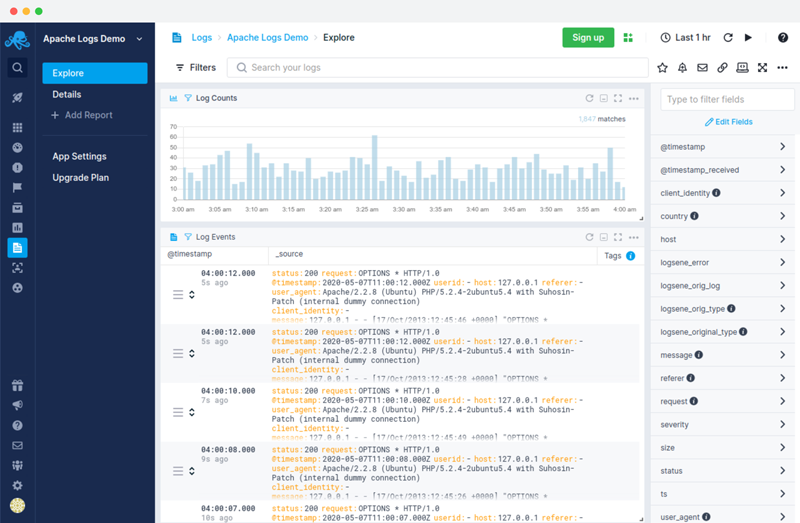


От панелей мониторинга до метрик, логов и трассировок.

Сильные стороны:

* Удобные дешборды
* Панели с симпатичными настраиваемыми графиками
* Аннотации для выделения важных деталей
* Работает с большим количеством движков баз данных.
* Имеет множество плагинов и легко совмещается с другими инструментами.
* OpenSource

Sematext Cloud[13]



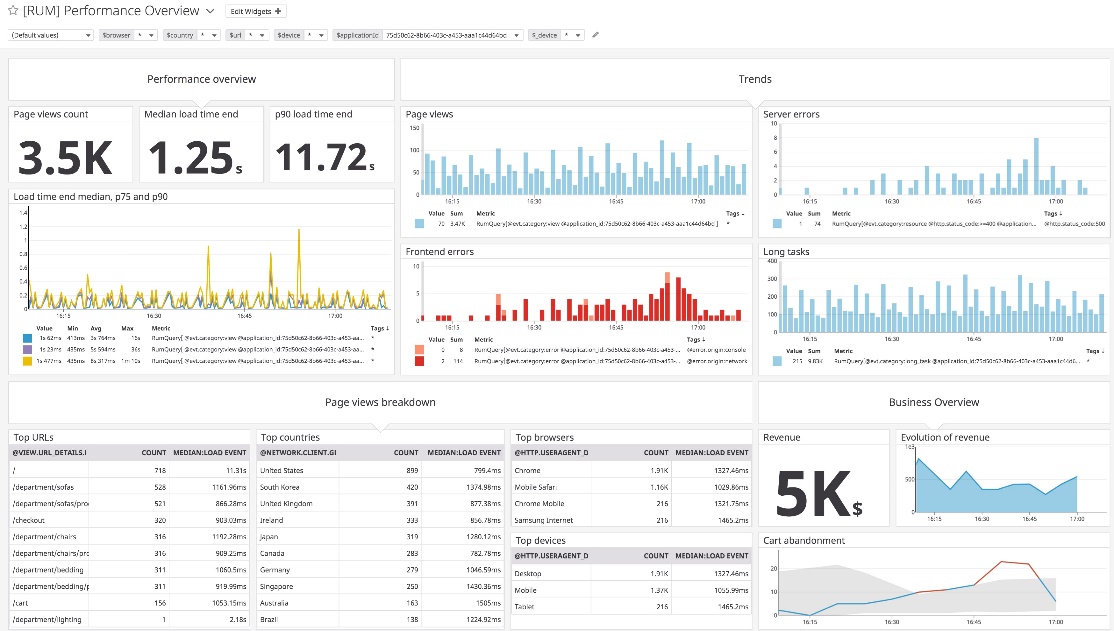
Облако Sematext и его локальная версия - Sematext Enterprise – это решение для мониторинга с полным стеком, которое легко настроить и получить полную картину вашей ИТ – инфраструктуры.

Автоматическое обнаружение служб Sematext позволяет автоматически начинать мониторинг непосредственно через пользовательский интерфейс. Есть мощная система оповещений с обнаружением аномалий и планированием. Sematext – это решение, которое обеспечивает как быстрый, так и прогнозирующий мониторинг с простым анализом.

Сильные стороны:

* Мониторинг сети, баз данных и процессов.
* Мощные возможности информационной панели для отображения практически любых данных, отправляемых в Sematext
* Мониторинг реальных и фиктивных пользователей для полной видимости того, как воспринимается ваш интерфейс и насколько
* быстры и работоспособны ваши API.
* API-интерфейсы Elasticsearch и InfluxDB позволяют интегрировать любые инструменты, которые работают с ними, такие, как
* Logstash, Filebeat, Fluentd, Logagent, Vector и т.д.

DataDog[14]



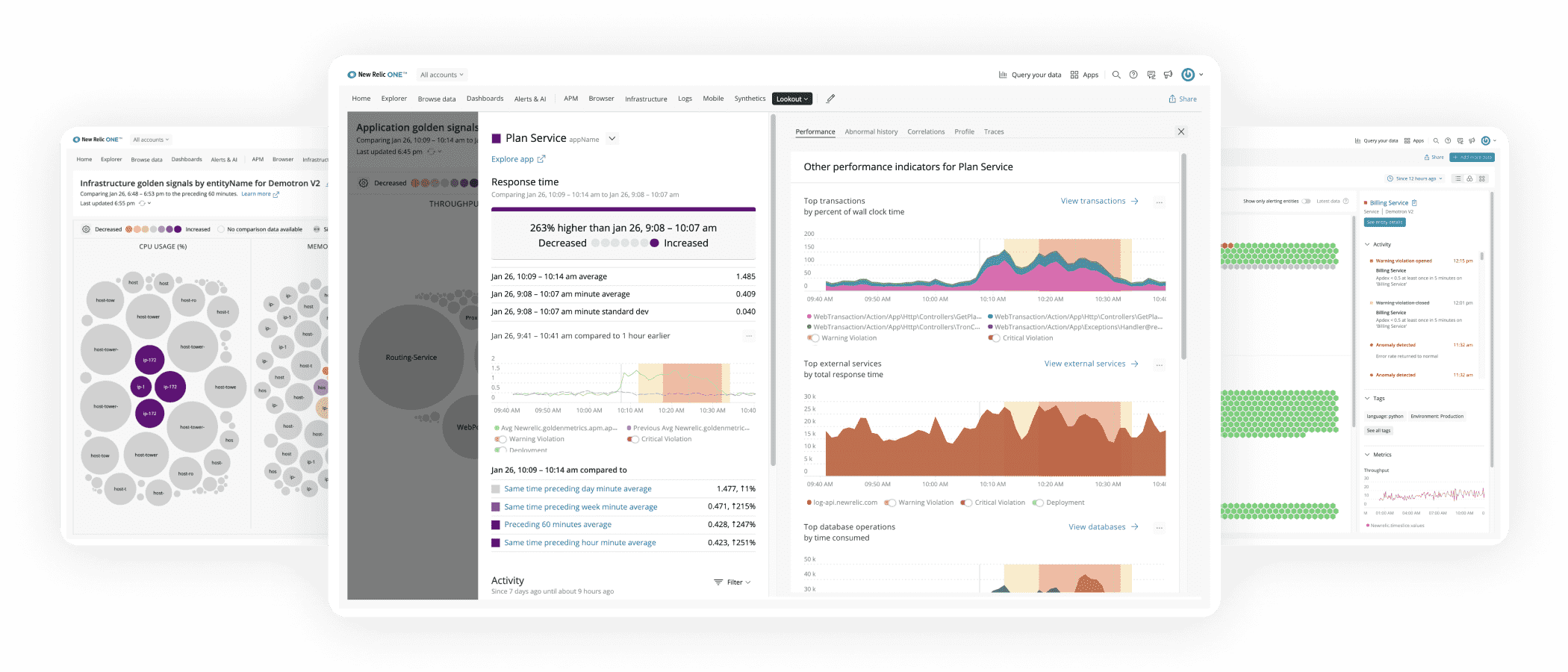
Datadog — это решение, предоставляющее большой набор функций, необходимых для мониторинга облачной инфраструктуры,

приложений, контейнеров, сети, журналов или даже бессерверных функций, таких как AWS lambdas. Однако за гибкость и функциональность приходится платить – установка агента на основе конфигурации может занять много времени (например, мониторинг процессов требует редактирования конфигурации и перезапуска), и может пройти довольно много времени, прежде чем вы начнете видеть все показатели, журналы и трассировки.

Плюсы:

* Полное решение для обеспечения наблюдаемости – метрики, журналы, безопасность все в одном.
* Мощные и очень гибкие функции анализа данных с предупреждениями и пользовательскими панелями мониторинга.
* Расширенная интеграция журналов, включая приложения, контейнеры, облачных провайдеров, клиентов и обычных сервисов.

New Relic[15]

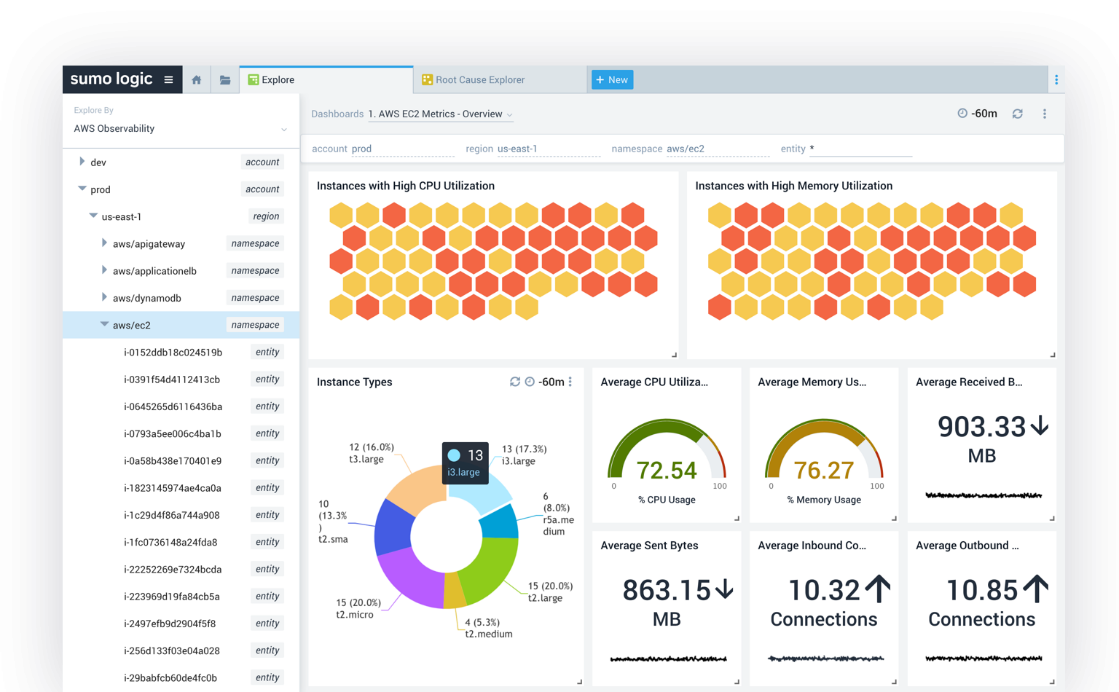


New Relic - так же решение предоставляющее широкий спектр возможностей мониторинга. Он включает мониторинг производительности мобильных, веб-и серверных приложений с широкой поддержкой информационных панелей, распределенной трассировки и логов. Несмотря на то, что требуется установка вручную, она надежна благодаря широкому спектру поддерживаемых языков программирования, что является большим преимуществом.

Плюсы:

* Интегрированное оповещение с обнаружением аномалий.
* Мониторинг производительности приложений с помощью панели мониторинга и поддержки часто используемых языков, включая
* C++.
* Интеграция с большинством известных облачных провайдеров, таких как AWS, Azure и облачная платформа Google.

Sumo Logic[16]



Sumo Logic - это система мониторинга с сильным акцентом на работу с логами. С помощью таких инструментов, как LogReduce и LogCompare, вы можете не только просматривать логи за определенный период времени, но и уменьшать объем данных, необходимых для анализа и сравнения периодов.

Плюсы:

* Удобный интерфейс, который не перегружает начинающих пользователей и по-прежнему удобен для опытных.
* Возможность уменьшить количество похожих логов во время чтения и сравнивать периоды времени вместе, что может помочь
* быстро обнаружить различия, аномалии и отследить проблемы.
* Доступен ограниченный бесплатный уровень, которого может быть достаточно для очень небольших компаний.

И другие инструменты:

* Zabbix
* Stackify Retrace
* Azure Monitor
* Zenoss
* BMC TrueSight Pulse
* Solarwinds
* Amazon CloudWatch
* PagerDuty
* VictorOps
* xMatters
* OpsGenie
* стек elastic ELK (Elasticsearch, LogStash и Kibana).

Множество существующих инструментов аналитики и визуализации работает с Elasticsearch MySql InfluxDB

Принцип работы у подобных программ в том, чтобы на каждом узле запустить процесс, логгирующий с состояние ВУ, и периодически посылать метрики серверу. Тот в свою очередь хранит большую базу данных, которую отдает в аналитику и визуализацию.

## Обзор инструментов реализации

Средства Python предоставляют решения вопроса сбора данных о ВУ, но не все из них кроссплатформерные. Модули, которые имеют отношения к сбору нужных метрик:

* PSI
* psutil
* os
* platform

Из перечисленных psutil кажется наиболее подходящим и дает доступ ко всем нужным данным, да и поддерживает множество ОС. К тому же у этого модуля есть достаточно подробная документация [6].

На роль базы данных для клиента была выбрана SQLite, т.к. данные будут локально храниться на ВУ. Для работы с ней есть средства в модуле sqlite3[7].

Для коммуникации с сервером есть множество вариантов. Было рассмотрено два из них.

Передача файлов по сокету, где мы просто пакуем наши метрики в json документ и отправляем на сервер по его запросу.

И второй поднятие приложения с get запросами с помощью fastAPI. Это приложение будет подключено к той же базе данных, в которой собираются наши метрики.

# Результаты

Были изучены основы языка Python и несколько дополнительных пакетов. После знакомства с языком и инструментами был проведен обзор существующих систем и корректировка набора выбранных инструментов реализации. После обсуждения с командой было установлено, что клиентский модуль должен обладать следующим функционалом:

* Сбор метрик вычислительного узла.
* Хранение метрик.
* Предоставлению серверу информации о состоянии ВУ по запросу, как о последнем состоянии, так и за промежуток времени.
* Возможность настраивать частоту сбора с помощью конфигурационного файла

Были реализованы все их этих функций с помощью следующих модулей: psutil, sqlite3, configparser, soket, fastAPI. Кроме того, было добавлено логгирование работы модуля средствами модуля logging. В итоге реализован модуль, собирающий данные о состоянии CPU, RAM, дисков и сети в базу данных sqlite. В одной вариации в двух потоках запускается сбор данных и ответы на запросы. В другой запускается отдельная программа, забирающая данные из локальной базы данных и отвечающая на запросы по парадигме REST API.

# Заключение

Во время прохождения практики были изучены и применены на практике основы языка Python и парадигма REST API. Так же были изучены подходы к мониторингу вычислительных систем.

# Информационные источники

1. Репозиторий проекта (<https://github.com/MAGGen-hub/Observability-Platform/>)
2. Бакалаврская программа «Прикладная математика и информатика» (<https://www.hse.ru/ba/se/>)
3. 20 Best Cloud Monitoring Tools & Services / Режим доступа: <https://sematext.com/blog/cloud-monitoring-tools/>, свободный.

(дата обращения: 4.07.2021).

1. What is Observability? | Sumo Logic / Режим доступа: <https://www.sumologic.com/glossary/observability/> свободный.

(дата обращения: 4.07.2021).

1. List of Top Observability Tools 2021/ Режим доступа: <https://www.trustradius.com/observability-tools> свободный.

(дата обращения: 4.07.2021).

1. Psutil documentation / Режим доступа: <https://psutil.readthedocs.io/en/latest/>, свободный. (дата обращения: 5.07.2021).
2. sqlite3 — DB-API 2.0 interface for SQLite databases — Python 3.9.6 documentation / Режим доступа: <https://docs.python.org/3/library/sqlite3.html>, свободный. (дата обращения: 5.07.2021).
3. FastAPI / Режим доступа: <https://fastapi.tiangolo.com/>, свободный.

(дата обращения: 5.07.2021).

1. Configuration file parser — Python 3.9.6 documentation / Режим доступа: <https://docs.python.org/3/library/configparser.html>, свободный.

(дата обращения: 7.07.2021).

1. socket — Low-level networking interface — Python 3.9.6 documentation / Режим доступа: <https://docs.python.org/3/library/socket.html>, свободный.

(дата обращения: 8.07.2021).

1. Программирование на Python. Принципы ООП. / Режим доступа: <https://smartiqa.ru/courses/python/lesson-6>, свободный.

(дата обращения: 5.07.2021).

1. Logging HOWTO — Python 3.9.6 / Режим доступа:  [https://docs.python.org/3/howto/logging.html](https://docs.python.org/3/library/socket.html), свободный.

(дата обращения: 10.07.2021).

1. Grafana® Features | Grafana Labs / Режим доступа: <https://grafana.com/grafana/> свободный. (дата обращения: 10.07.2021).
2. Sematext Cloud: Performance Monitoring & Log Management Tools - Sematext / Режим доступа: <https://sematext.com/cloud/> свободный. (дата обращения: 10.07.2021).
3. Cloud Monitoring as a Service | Datadog / Режим доступа: https://www.datadoghq.com/ свободный. (дата обращения: 10.07.2021).
4. New Relic | Deliver more perfect software / Режим доступа: <https://newrelic.com/>, свободный. (дата обращения: 10.07.2021).
5. Cloud Log Management, Monitoring, SIEM Tools | Sumo Logic / Режим доступа: <https://www.sumologic.com/>, свободный. (дата обращения: 10.07.2021).

# Приложение А.

Примеры запросов через fastAPI.