

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

ОТЧЕТ
по производственной практике
Тема: Практическое применение облачных технологий

Студенты гр. 9309

Аль Сайед А.З.

Серов А.В.

Юшин Е.В.

Преподаватель

Горячев А.В.

Санкт-Петербург

2022

ЗАДАНИЕ
НА «НАИМЕНОВАНИЕ ПРАКТИКИ» ПРАКТИКУ

Студенты гр. 9309: Аль Сайед А.З., Серов А.В., Юшин Е.В.

Группа 9309

Тема практики: Практическое применение облачных технологий

Сроки прохождения практики: 29.06.2022 – 12.07.2022

Дата сдачи отчета: 12.07.2022

Дата защиты отчета: 12.07.2022

Студенты гр. 9309

Аль Сайед А.З.

Серов А.В.

Юшин Е.В.

Преподаватель

Горячев А.В.

Аннотация

В данной практической работе мы используем облачные технологии Yandex.Cloud для создания Telegram-бота, который отзеркаливает отправленные ему сообщения написанного на языке программирования JavaScript с использованием Yandex Cloud Functions и Yandex Object Storage с возможностью отслеживания сессии, мониторинга запросов к боту, а также просмотру логов.

Summary

In this practical work, we use Yandex.Cloud cloud technologies to create a Telegram bot that mirrors messages sent to it written in the JavaScript programming language using Yandex Cloud Functions and Yandex Object Storage with the ability to track the session, monitor requests to the bot, and view logs.

Содержание

Аннотация	3
Summary	3
1.Создание бота	5
1.2 Создание функции	6
1.3 Обзор функции	7
1.4 Редактор кода	8
1.5 Создание бакета	9
1.6 Установка вебхука	9
1.8 Пример работы бота	11
1.9 Дополнительные возможности сервиса	12
Заключение	15

1. Создание бота

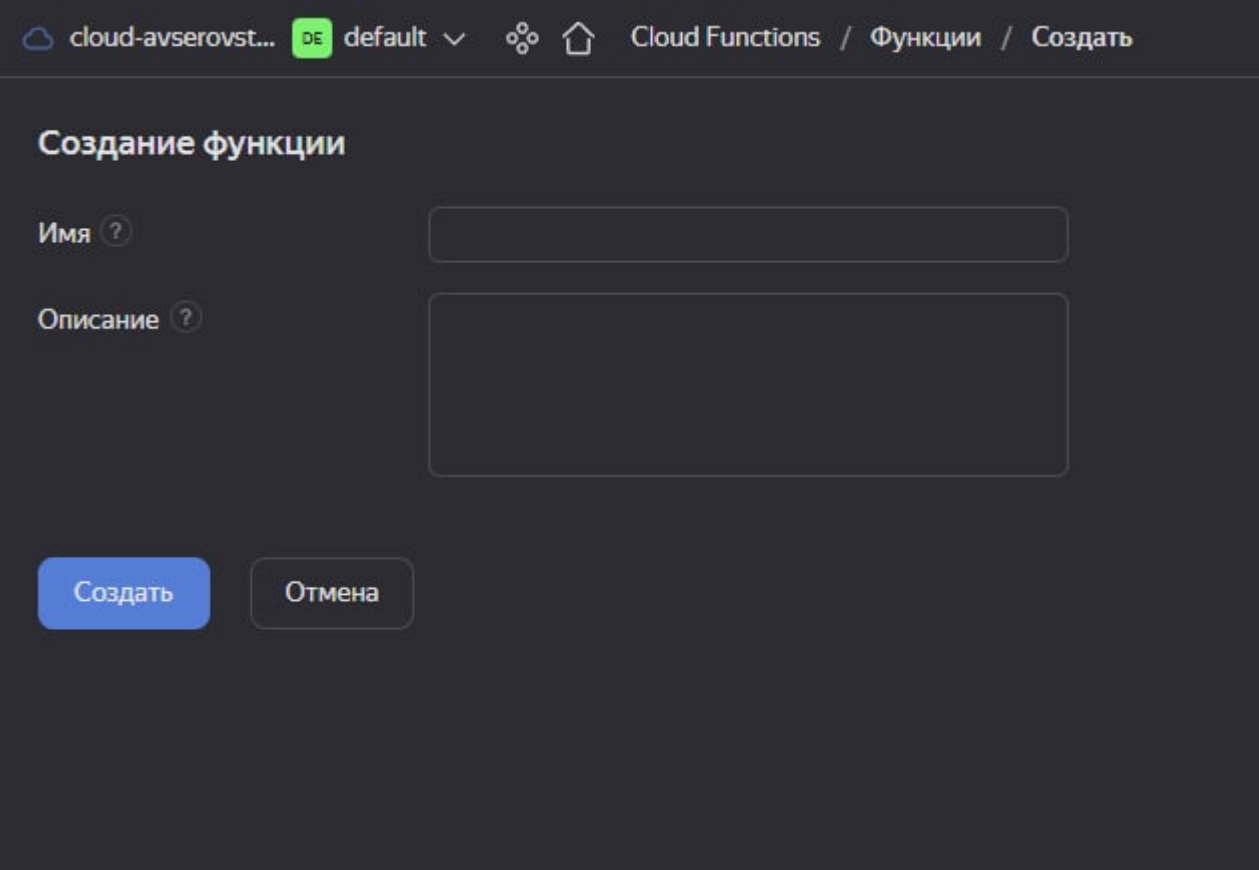
Для создания Telegram-бота необходимо открыть само приложение, найти @BotFather и вызвать команду /newbot, позволяющую создать нового бота. Назовём его «practice» и дадим ему username – «practice9309bot». После успешного создания уникального имени бота, нам надо скопировать и сохранить токен, присутствующий на скриншоте. С его помощью мы установим исходящий из Телеграма вебхук на нашу функцию.



Рисунок 1 – создание бота

1.2 Создание функции

Следующим шагом будет создание функции в сервисе Yandex Cloud Functions. Здесь мы должны прописать имя бота и опционально дать описание. Назовём нашего бота «tg-bot-practice»



The screenshot shows the 'Создание функции' (Create Function) page in the Yandex Cloud console. The breadcrumb navigation at the top reads 'Cloud Functions / Функции / Создать'. The page has a dark theme. It contains two input fields: 'Имя' (Name) with a question mark icon and 'Описание' (Description) with a question mark icon. At the bottom, there are two buttons: 'Создать' (Create) in blue and 'Отмена' (Cancel) in grey.

Рисунок 2 – создание функции

1.3 Обзор функции

После успешного создания функции, новая функция появится в списке. Нажимаем на неё – это приведёт нас на страницу «Обзор» нашей функции. Здесь надо включить переключатель «Публичная функция», чтобы она стала доступна из внешнего (Для Яндекс.Облака) мира, а значение полей «Ссылка для вызова» и «Идентификатор» хранить в тайне, чтобы функцию не могли использовать мошенники.

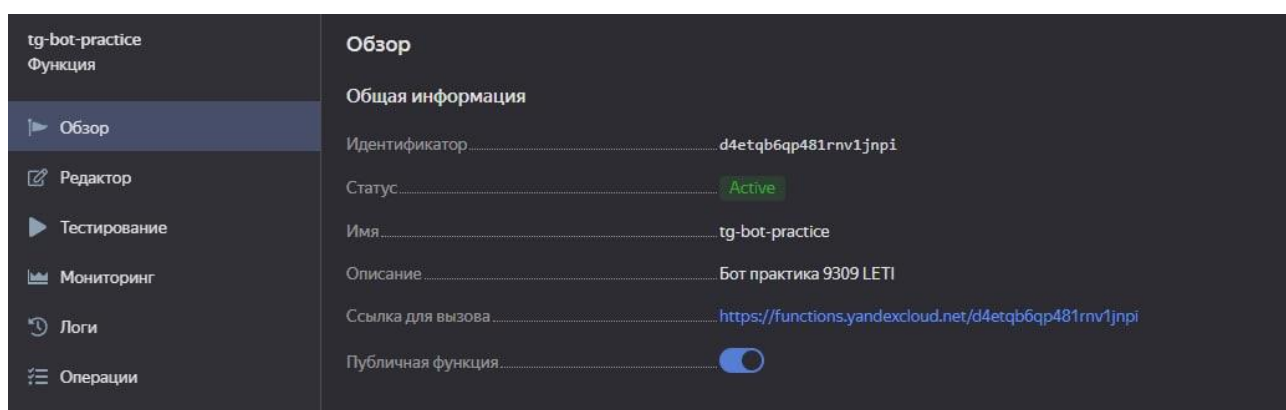


Рисунок 3 – Обзор функции

1.4 Редактор кода

Далее переходим во вкладку «Редактор», где мы располагаем исходный код бота. Чуть ниже указываем Точку входа – `index.bot`, где `index` это имя файла(`index.js`), а `bot` – имя функции (`module.exports.bot`). Все остальные поля оставляем по умолчанию и создаём версию.

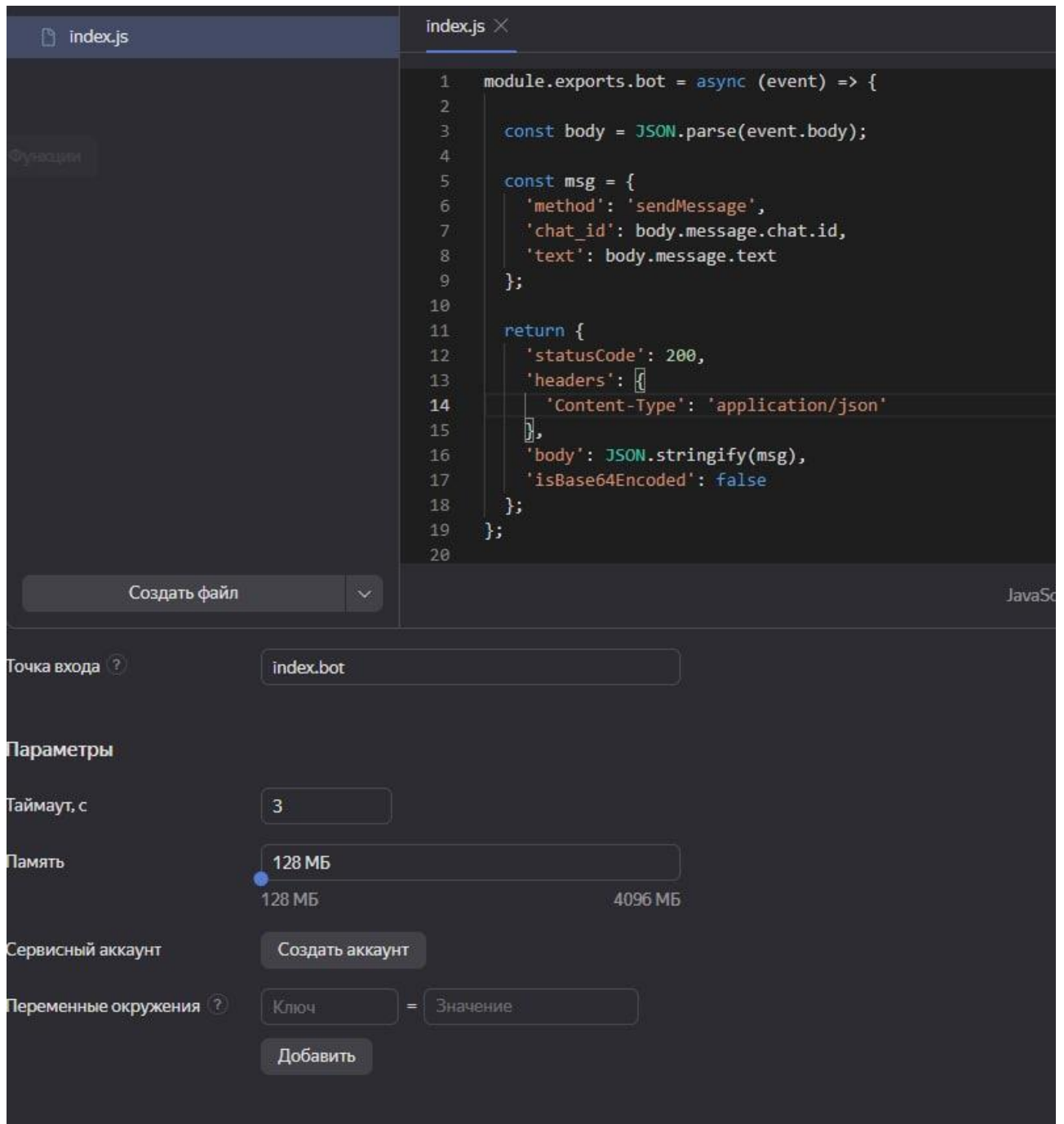
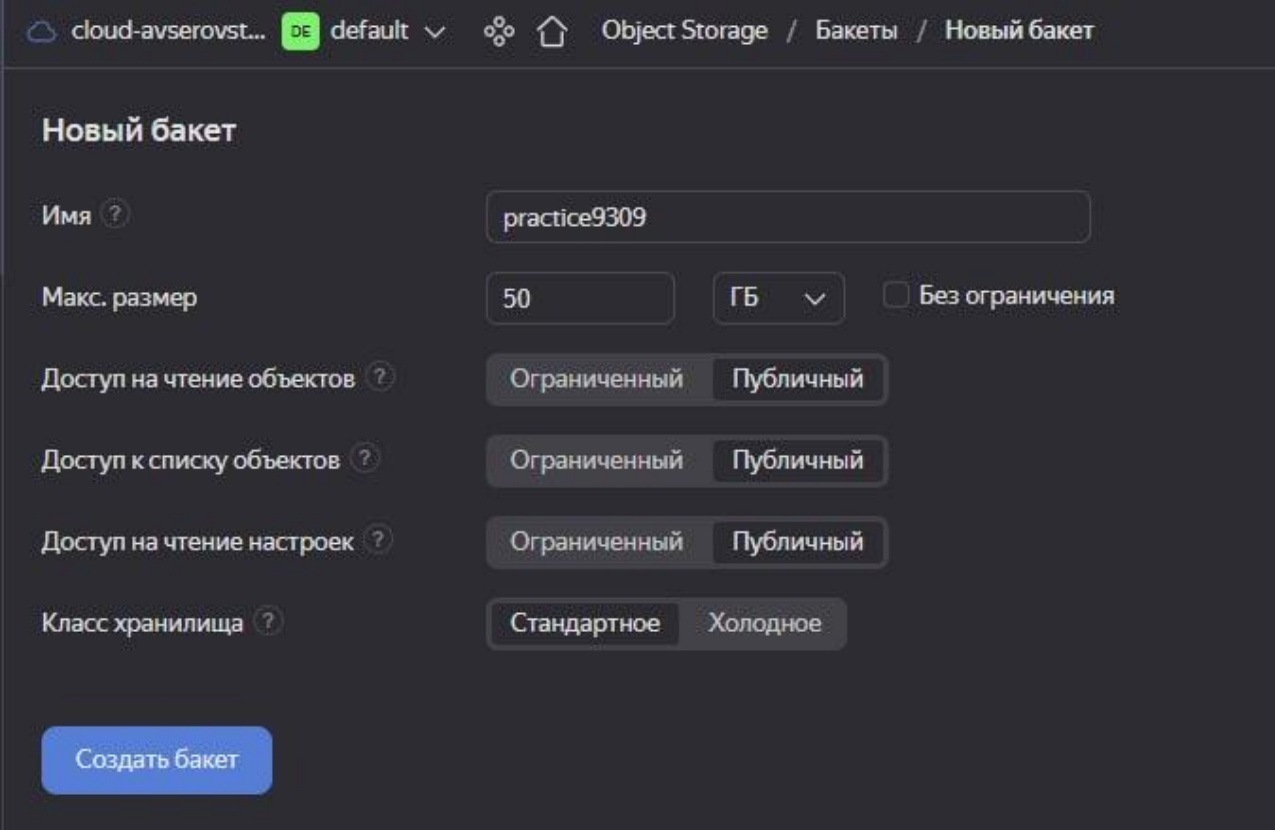


Рисунок 4 – Редактор кода

1.5 Создание бакета

Теперь, когда у нас создана Яндекс-функция, создадим ещё и бакет для хранения файлов изображений. Выбираем пункт «Object Storage», нажимаем кнопку «Создать бакет», даём ему название «practice9309» и доступ на чтение объектов делаем публичным, все остальные поля оставляем без изменений и завершаем создание бакета.



cloud-avserovst... default Object Storage / Бакеты / Новый бакет

Новый бакет

Имя [?] practice9309

Макс. размер 50 ГБ ☐ Без ограничения

Доступ на чтение объектов [?] Ограниченный Публичный

Доступ к списку объектов [?] Ограниченный Публичный

Доступ на чтение настроек [?] Ограниченный Публичный

Класс хранилища [?] Стандартное Холодное

Создать бакет

Рисунок 5 – создание бакета.

1.6 Установка вебхука

Следующим шагом будет установка вебхука. То есть, когда бот получит обновление (например, сообщение от пользователя), с сервера Telegram в нашу Яндекс-функцию будет отправлен запрос с данными.

`https://api.telegram.org/bot{bot_token}/setWebHook?url={webhook_url}` –

данную строку необходимо вставить в адресное поле браузера, а затем обновить страницу, предварительно заменив {bot_token} на токен, полученный нами при создании нашего бота, а {webhook_url} – на URL нашей Яндекс-функции.

```
{ "ok": true, "result": true, "description": "Webhook was set" }
```

Рисунок 6 – результат отправки вебхука.

1.8 Пример работы бота

После данных процедур бот должен успешно работать. Бот должен «отзеркаливать» сообщения, отправленные ему. Проверяем.

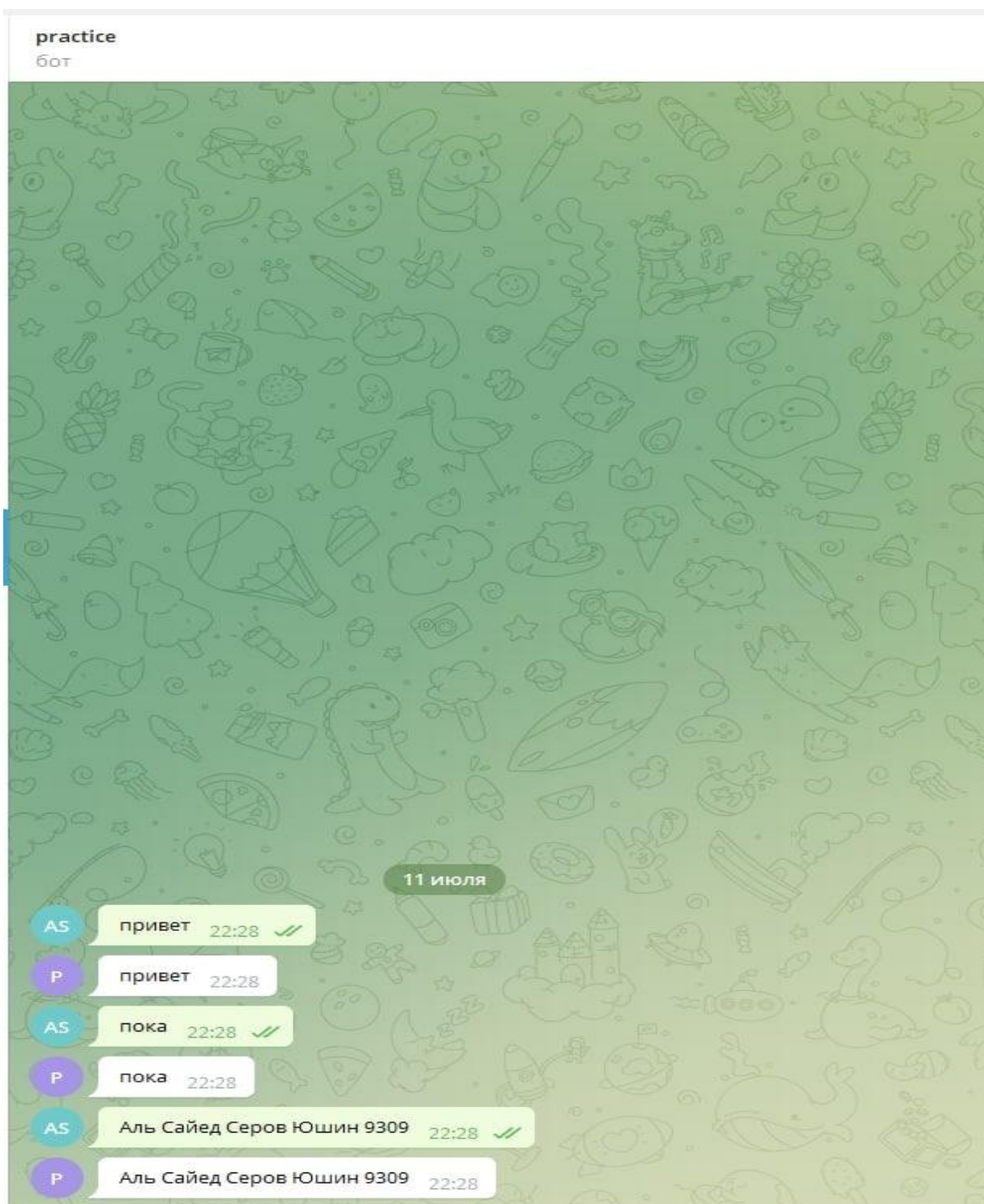


Рисунок 7 – проверка бота.

1.9 Дополнительные возможности сервиса

Нам также доступны такие разделы как: «Тестирование», «Мониторинг», «Логи», «Операции». Рассмотрим их по порядку.

Тестирование – при тестировании данным способом выполняется HTTP-запрос с методом POST и параметром `integration=raw`.

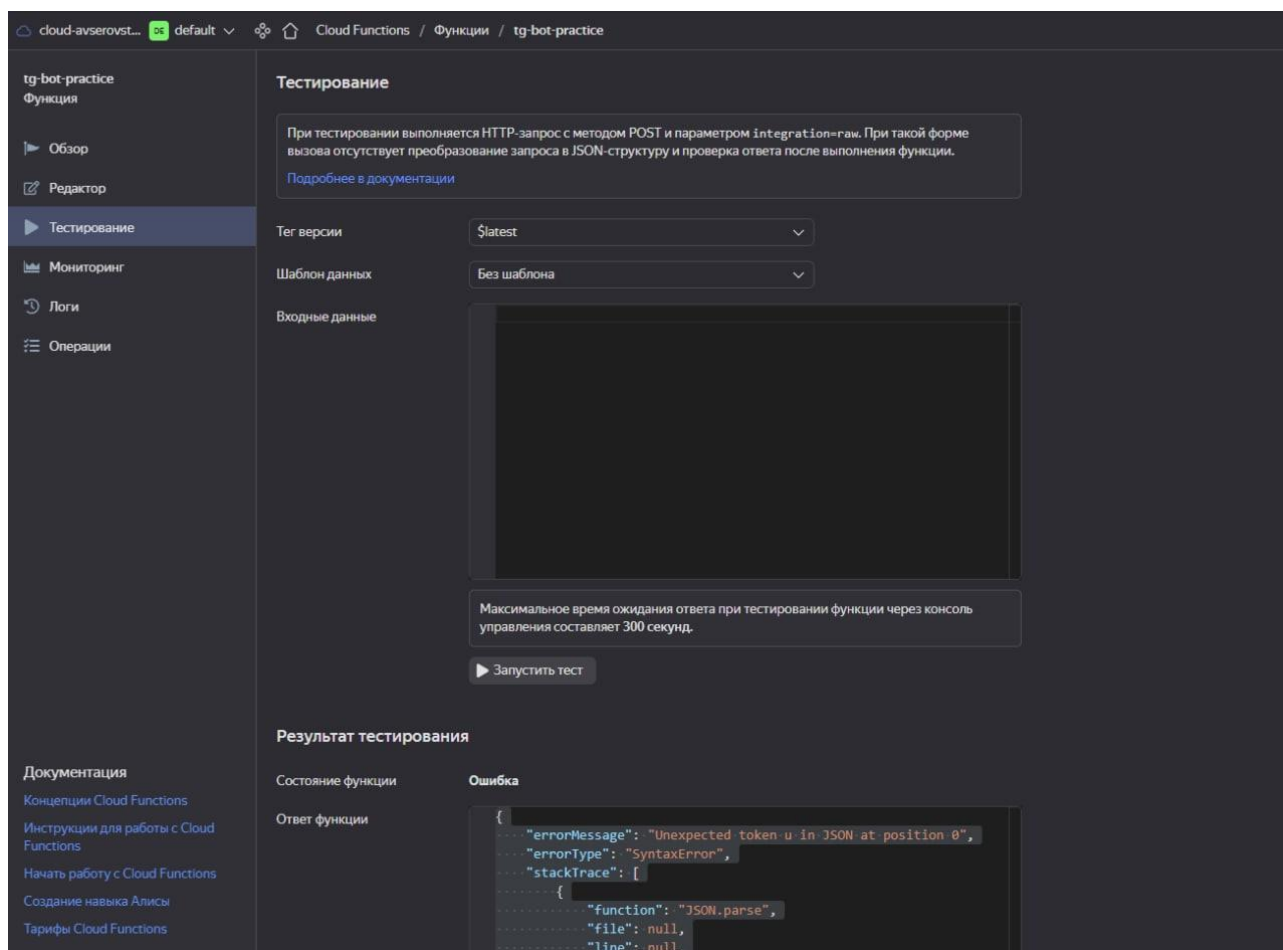


Рисунок 8 – Тестирование

Мониторинг позволяет отслеживать работу функций с помощью инструментов в консоли управления. Данные инструменты предоставляют диагностическую информацию в виде графиков. Собирает значения метрик и отображает графики сервис Yandex.Monitoring.



Рисунок 9 – мониторинг.

Помимо мониторинга, мы также можем отслеживать логи нашей функции.

Хранятся данные логи 3 суток, также можно указать период, за который мы хотим смотреть логи. По умолчанию это 1 час.

cloud-avserovst... default Cloud Functions / Функции / tg-bot-practice			
<div> <div>tg-bot-practice</div> <div>Функция</div> </div> <div> <div>Обзор</div> <div>Редактор</div> <div>Тестирование</div> <div>Мониторинг</div> <div>Логи</div> <div>Операции</div> </div> <div> <div>Документация</div> <div>Концепции Cloud Functions</div> <div>Инструкции для работы с Cloud Functions</div> <div>Начать работу с Cloud Functions</div> </div>			
<div> <div>Запрос</div> <div>Уровень: Все</div> <div>Кол-во: 100</div> <div>1h 3h 1d 1w 2w Другое</div> <div>2022-07-11 16:59 – 17:59</div> </div> <div> <div>Время (UTC+3)</div> <div>Уровень</div> <div>JSON</div> <div>Сообщение</div> </div> <div>Загрузить 100 предыдущих записей</div>			
17:52:14, 11.07.2022	ERROR		[ERROR] this.handlerFn is not a function
17:52:14, 11.07.2022			Stack at vp-<anonymous> (/function/runtime/bootstrap262:116)
17:52:14, 11.07.2022			at Generator.throw (<anonymous>)
17:52:14, 11.07.2022	INFO		END RequestID: 763c8b41-0b56-4083-b6c4-0786c5f529b2
17:52:14, 11.07.2022	ERROR		ERROR RequestID: 763c8b41-0b56-4083-b6c4-0786c5f529b2 Code: 502 Message: Error during function invocation
17:52:14, 11.07.2022			[{"errorMessage":"Cannot find module 'node-fetch'\nRequire stack:\n- /function/code/index.js","errorType":"Error","stackTrace":[{"line":"","function":"","functionName":"","column":"","fileName":"","methodName":"","methodNamePrefix":"","methodNameSuffix":"","methodNameSuffixPrefix":"","methodNameSuffixSuffix":"","methodNameSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixPrefix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffix":"","methodNameSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixSuffixPrefix","
17:52:56, 11.07.2022	INFO		START RequestID: cc84d784-455d-4e33-936a-8071962b049c Version: d4empj678tp1vbnunf
17:52:56, 11.07.2022	ERROR		[ERROR] Cannot find module 'node-fetch'
17:52:56, 11.07.2022			Require stack:
17:52:56, 11.07.2022			- /function/code/index.js
17:52:56, 11.07.2022			Stack: Require stack:
17:52:56, 11.07.2022			- /function/code/index.js
17:52:56, 11.07.2022			at Function.Module.resolveFilename (node:internal/modules/cjs/loader:933:15)
17:52:56, 11.07.2022			at Function.Module.load (node:internal/modules/cjs/loader:778:27)
17:52:56, 11.07.2022			at Module.require (node:internal/modules/cjs/loader:1005:19)

Рисунок 10 – логи

В разделе «операции» нам доступны все операции, которые были произведены над функцией. Показано, кто произвёл операцию, её текущий статус, описание, а также идентификатор.

tg-bot practice
Функция

Обзор

Редактор

Тестирование

Мониторинг

Логи

Операции

Операция

Статус: Не выбран ▾

Идентификатор	Статус:	Пользователь	Описание операции
d4e39d6fz7m4qj3l6dp	✓ Завершена	avserov@studetu.ru	Удалить сессию
d4eslmvQpbshJ16R51	✓ Завершена	avserov@studetu.ru	Создать сессию
d4esldr1ScrbMiv7o	✓ Завершена	avserov@studetu.ru	operation_deploy_session; logs https://storage.yandexcloud.net/build-logs/b1gfbzogn014/jctfd/54etqb6gp481mr1jpvi/54ec3c8bmch0usdownoutput?X-Amz-Algorithm=AWS4-HMAC-SHA-
d4efumljgc3IeF917gd	✓ Завершено	avserov@studetu.ru	Создать сессию
d4efon7wmgdhvakf9be	✓ Завершена	avserov@studetu.ru	operation_deploy_session; logs https://storage.yandexcloud.net/build-logs/b1gfbzogn014/jctfd/54etqb6gp481mr1jpvi/54emploj7tp1vpbnuf.output?X-Amz-Algorithm=AWS4-HMAC-SH
d4eqka3t1d7gl1lkkr-j	✓ Завершена	avserov@studetu.ru	Загрузить файл
d4ekiqb1SafZ14tkB391	✓ Завершена	avserov@studetu.ru	Создать сессию
d4egoVvsq85sknlpzg	✓ Завершена	avserov@studetu.ru	Удалить сессию
d4elkv1k7neeporfjsht7j	✓ Завершена	avserov@studetu.ru	Создать сессию
d4ejyudfj2jjtmzhma5	✓ Завершена	avserov@studetu.ru	Удалить сессию
d4edho6bzNIdp88le7	✓ Завершена	avserov@studetu.ru	Создать сессию
d4ekt17o7ebvdop3K58	✓ Завершена	avserov@studetu.ru	operation_deploy_session; logs https://storage.yandexcloud.net/build-logs/b1gfbzogn014/jctfd/54etqb6gp481mr1jpvi/54e7s752cz7fkx85s.output?X-Amz-Algorithm=AWS4-HMAC-SHA
d4edlsbsj5sqgadmbv9	✓ Завершена	avserov@studetu.ru	Загрузить файл
d4edszkIsac1cdF7avtt	✓ Завершена	avserov@studetu.ru	Загрузить файл
d4e5bjvc2kcJoFF1qils	✓ Завершена	avserov@studetu.ru	Загрузить файл
d4evbt15jhmk437aptn	✓ Завершена	avserov@studetu.ru	Загрузить файл
d4eksln175v9H6dezvq	✓ Завершена	avserov@studetu.ru	Загрузить файл

Документация

Концепция Cloud Functions

Инструкция для работы с Cloud Functions

Начать работу с Cloud Functions

Рисунок 11 – операции функции.

Заключение

Во время прохождения производственной практики нами были реализованы различные функции Yandex.Cloud, которые ранее были изучены при выполнении НИР. Так же были получены практические навыки по созданию Телеграмм бота и связи его через вебхук с основной функцией, которая лежала на Yandex.Cloud.