

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**ОТЧЕТ о выполнении  
ЛАБОРАТОРНОЙ РАБОТЫ №1  
по дисциплине “Основы облачных технологий”**

Работу выполнил студент 4 курса 4ИТ гр. \_\_\_\_\_ Молчанов И.А.

Работу проверил канд. техн. наук, доц. \_\_\_\_\_ Полупанов А. А.

Краснодар 2023

Тема: Разработка веб-сайта на облачной платформе Yandex Cloud для ввода и хранения заметок.

Цель: Изучение теоретических основ облачных вычислений, внутренней структуры и практической реализации, и прикладных примеров использования облачных вычислений и веб-сервисов.

Задание: Разработка облачного приложения с использованием публичных веб-сервисов.

Постановка задачи: разработать веб-сайт на облачной платформе Yandex Cloud для ввода и хранения заметок.

## **1 Аккаунт Yandex Cloud**

Платформа Yandex Cloud имеет несколько пользовательских интерфейсов, например, консоль управления и интерфейс командной строки. Для доступа к любому пользовательскому интерфейсу потребуется пользовательский аккаунт.

Платежный аккаунт используется для идентификации пользователя, оплачивающего ресурсы Yandex Cloud. Даже если вы планируете пользоваться только бесплатными сервисами, вам все равно необходим платежный аккаунт: на него начисляются гранты и промокоды.

Для активации платежного аккаунта необходимо привязать банковскую карту, чтобы убедиться, что вы человек, а не робот. При привязке карты будет списана и сразу вернется небольшая сумма. Так проверяется, что карта настоящая.

Средства с карты для оплаты сервисов и ресурсов Yandex Cloud будут списывать только после того, как пользователь явно разрешит это делать, то есть перейдет на платное потребление.

### **Создание платежного аккаунта**

Платежный аккаунт необходим, даже если планируется пользоваться только бесплатными сервисами. При создании первого платежного аккаунта, привязанного к пользовательскому аккаунту, будет начислен стартовый грант.

Необходимо авторизоваться с помощью Яндекс ID и войти в консоль управления. Подтвердить создание своего первого облака и перейти к созданию платежного аккаунта:

1. В консоли управления перейти в раздел Биллинг.

2. На странице *Аккаунты* нажать кнопку *Создать аккаунт*. Заполнить данные.

3. Если в блоке *Плательщики* показан список доступных плательщиков, можно выбрать одного из них или добавить нового.

4. Для добавления нового плательщика необходимо выбрать *Тип плательщика*:

5. Привязать карту.

Для проверки валидности карты на счете будет заблокирована незначительная сумма. Эти средства не списываются и будут снова доступны после завершения проверки и разблокировки. Срок разблокировки несписанных средств зависит от условий работы вашего банка.

Если это первый платежный аккаунт в Yandex Cloud, то будет доступно подключение пробного периода.

- После завершения пробного периода ресурсы будут приостановлены. Для возобновления работы потребуется перейти на платную версию.

- Если не подключать пробный период на данном этапе, аккаунт будет создан с платным потреблением: после использования стартового гранта не придется переходить на платную версию.

6. Нажать кнопку *Создать*.

В рамках одного пользовательского аккаунта все плательщики должны быть либо резидентами РФ, либо резидентами РК, либо резидентами других стран.

## **2 Работа с Yandex Resource Manager**

Сервис для управления ресурсами Yandex Cloud — облаками, каталогами и другими ресурсами облачных сервисов.

Когда появляется доступ к Yandex Cloud, выделяется отдельное рабочее пространство — *облако*. В нем создаются каталоги.

В каталогах размещаются ресурсы — виртуальные машины, диски и др. При создании ресурса указывается, в каком каталоге он будет создан. Вложенные каталоги сейчас не поддерживаются.

Resource Manager предоставляет стандартную ресурсную модель. Эта модель используется в большинстве сервисов Yandex Cloud.

Все ресурсы в облаке по умолчанию изолированы от пользователей вне облака. Владелец облака может управлять правами доступа к облаку и ресурсам в нем.

Права доступа на ресурсы в облаке наследуются. Права на облако распространяются на все ресурсы внутри облака. Права на каталог распространяются на все ресурсы в каталоге.

Некоторые типы ресурсов не создаются в каталогах, поэтому имеют собственную логику проверки прав доступа. Например, при управлении ключами доступа для сервисного аккаунта проверяются права доступа на этот сервисный аккаунт.

*Облако* — это изолированное пространство, в котором создаются каталоги.

При создании ресурса внутри облака доступ к ресурсу закрыт для всех, кроме участников и владельцев облака.

## **Создание нового облака**

1. При регистрации будет автоматически создано облако с именем `cloud-  
<Яндекс ID>`. После того, как привязывается платежный аккаунт, появляется возможность создать дополнительное облако. На странице биллинга можно убедиться, что платежный аккаунт находится в статусе `ACTIVE` или `TRIAL_ACTIVE`. Если платежного аккаунта нет, также можно попросите администратора назначить роль `billing.accounts.member` на платежный аккаунт.

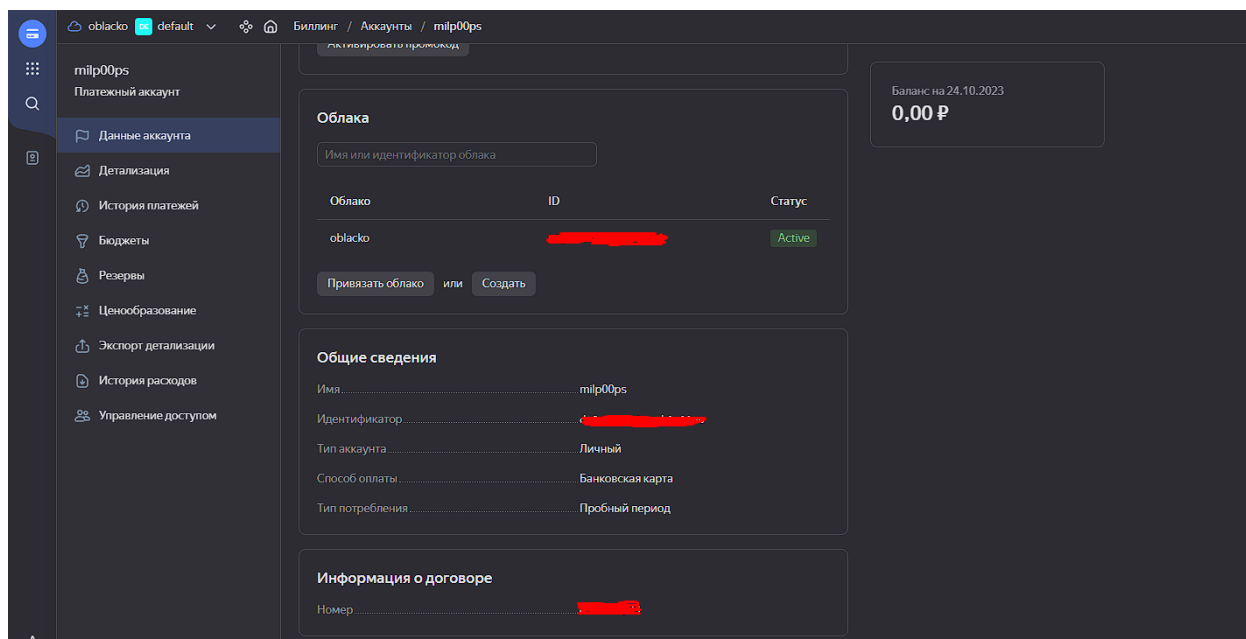


Рисунок 1 — статус Active

2. На стартовой странице консоли управления необходимо нажать значок напротив своего биллинг-аккаунта.

3. Выбрать *Создать облако*.

4. Ввести имя облака.

5. Указать организацию, к которой будет привязано облако. Если еще нет организации, она будет создана автоматически.

6. Нажать *Создать*. Произойдет перенаправление на страницу каталога `default` в новом облаке.

При создании облака автоматически получается роль владельца в этом облаке.

Владелец может создавать в этом облаке новые каталоги и ресурсы, а также управлять правами доступа к ним.

Каталог — это пространство, в котором создаются и группируются ресурсы Yandex Cloud.

Как и каталоги в файловой системе, каталоги в Yandex Cloud упрощают управление ресурсами. Можно группировать ресурсы в каталоги по типу

ресурса, по проекту, по отделу, который работает с этими ресурсами, или по любому другому признаку.

Так же можно управлять правами доступа сразу ко всем ресурсам в каталоге. Допустим, в организации есть сотрудники, которые работают только с виртуальными машинами. Тогда есть возможность создать каталог, в котором будут только виртуальные машины, и предоставить сотрудникам доступ к этому каталогу.

Посмотреть содержимое каталога в консоли управления тоже можно. Если необходимо узнать, к какому каталогу относится ресурс, можно воспользоваться API или CLI, выполнив метод Get для этого ресурса (команду get в CLI).

### **Создание каталога**

1. В консоли управления выбрать облако и нажать кнопку **+ Создать каталог**.
2. Ввести имя каталога.
3. (опционально) Ввести описание каталога.
4. Выбрать опцию *Создать сеть по умолчанию*. Будет создана сеть с подсетями в каждой зоне доступности. Также в этой сети будет создана группа безопасности по умолчанию, внутри которой весь сетевой трафик разрешен.
5. Нажать кнопку *Создать*.

### **3 Работа в редакторе кода**

Когда появилось отдельное рабочее пространство — *облако*, создаём в нем каталог. В каталоге размещаем ресурсы.

1. В консоли управления выбираем сервис *Cloud Functions*.
2. Выбираем функцию и переходим на вкладку *Редактор*.

3. Если функция пустая, то выбираем среду выполнения и нажимаем кнопку *Продолжить*.

4. В блоке *Код функции*:

- Выбираем среду выполнения, в нашем случае Node.js/18.
- Выбираем способ редактирования функции: Редактор кода.
- В панели навигации выбираем существующий файл или нажимаем кнопку Создать файл.
- В открывшемся окне вводим имя файла с расширением и нажимаем Создать.
- Указываем точку входа — имя функции, которая будет вызываться в качестве обработчика.

Формат: <имя файла без расширения>.<имя обработчика>.

Например, `index.myFunction`.

5. В блоке *Параметры*:

- Указываем таймаут.
- Указываем объем памяти.
- (опционально) Выбираем или создаём сервисный аккаунт. При добавлении сервисного аккаунта для функции можно получать его IAM-токен из контекста функции.
- (опционально) Добавляем переменные окружения.

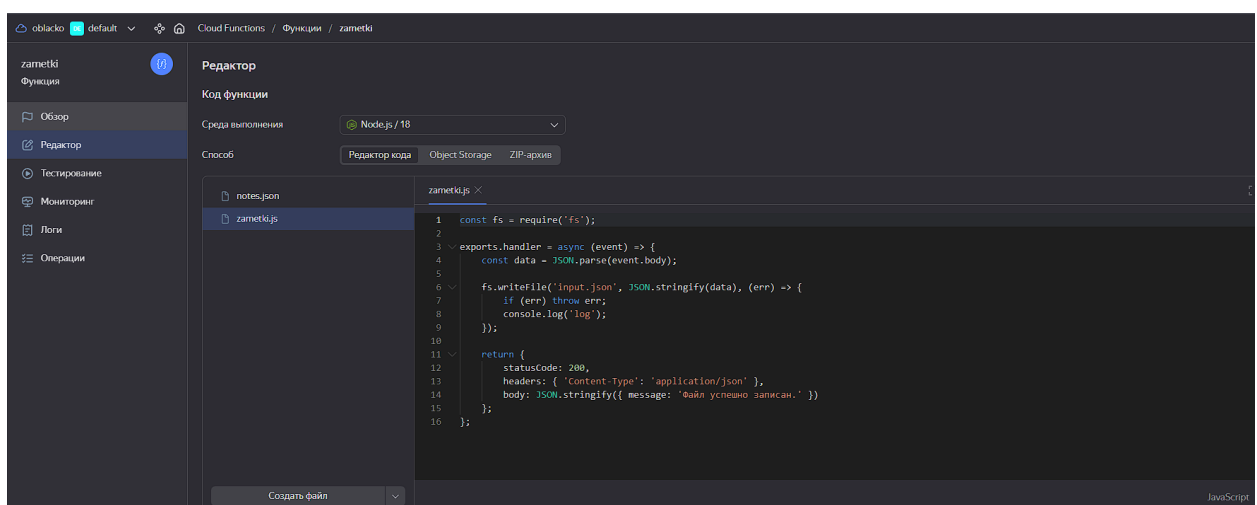


Рисунок 2 – Редактор кода Yandex Console



Параметры

Таймаут, с\*

Память   
128 МБ 4096 МБ

Сервисный аккаунт

Сеть  ▾

Переменные окружения ?  =

Секреты Yandex Lockbox  =  ▾  
 ▾  
 ▾

Рисунок 4 – Параметры

В файле `zametki.js` происходит передача данных о заметках в файл `json`.

6. В правом верхнем углу нажимаем *Создать версию*.  
Формируется ссылка для вызова.

**Обзор**

Идентификатор.....

Имя.....zametki

Статус.....Active

Дата создания.....24.10.2023, в 10:06

Ссылка для вызова.....<https://functions.yandexcloud.net/>

Публичная функция.....☐

**Последняя версия**

Дата создания.....24.10.2023, в 10:27

Таймаут, с.....1

Среда выполнения.....nodejs18

Точка входа.....script.sendDataToFunction

Память.....128 МБ

**Логирование**

Назначение.....Каталог

Каталог.....

Минимальный уровень логирования.....Не выбрано

Открыть в редакторе

Рисунок 5 – создание версии

В дальнейшем используем эту ссылку в мобильном приложении для получения данных.

## Заметки

Название:

Содержание:

Сохранить заметку

Название	Содержание
----------	------------

Рисунок 6 – основной экран веб-сайта

## Вывод

Рассмотрел облачную платформу Yandex Cloud. Изучил теоретическую основу и выявил особенности внутренней структуры. Разработал прикладные примеры использования.

## **Приложение 1**

Веб-сайт:

[https://github.com/IlyaMolchanovv/Yandex\\_Cloud\\_OOT](https://github.com/IlyaMolchanovv/Yandex_Cloud_OOT)