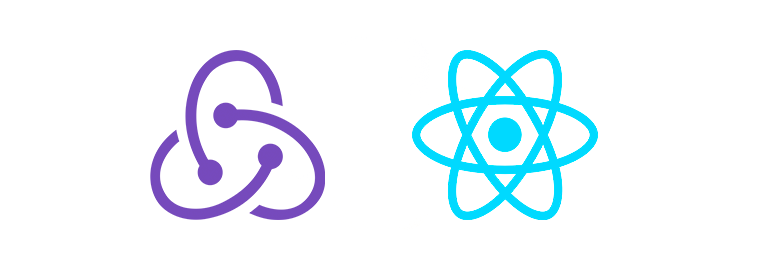
Как задеплоить веб-приложение на связке React и Redux за 10 минут

* [Статьи](https://tproger.ru/category/articles/), 30 июля 2017 в 20:55
* 5776



Перевод статьи [«How to deploy a live ReactJS/Redux website in under 10 minutes»](https://hackernoon.com/how-to-deploy-a-live-reactjs-redux-website-in-under-10-minutes-cadf73cfc75a" \t "_blank)

Если вы ищете способ быстро продемонстрировать коллегам или заказчикам идею своего веб-приложения, развернув его на сервере, эта статья для вас.

Вы получите аналогичное [~~приложение~~](http://onederful-quickstart.s3-website-us-west-2.amazonaws.com/) после выполнения этих действий:

* локальная установка ReactJS при помощи шаблона;
* настройка «корзины» AWS (Amazon Web Services) S3;
* создание учетных данных пользователя AWS для загрузки файлов на S3;
* развертывание шаблона на AWS;
* проверка работоспособности.

Необходимые инструменты:

* [Node.js](https://nodejs.org/en/download/) версии 6.0 или выше;
* [Yarn](https://yarnpkg.com/lang/en/docs/install/);
* Аккаунт [AWS](https://aws.amazon.com/) (бесплатного будет достаточно). Для интересующихся у нас есть [шпаргалка по веб-сервисам Amazon](https://tproger.ru/translations/aws-in-plain-russian/).

1. Установка ReactJS

Клонируем шаблон (запустите команду в терминале), заменив «NameOfApp» на имя своего приложения:

$ git clone -o onederful-quickstart -b master --single-branch \

https://github.com/alxyee/onederful-quickstart.git NameOfApp

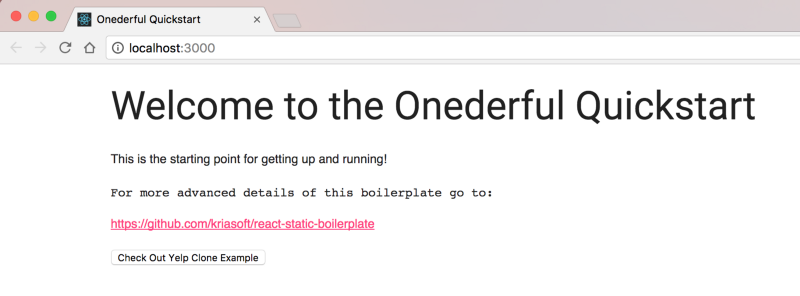
$ cd NameOfApp

Устанавливаем все библиотеки:

$ yarn install

Запускаем React по локальному адресу [~~http://localhost:3000/~~](http://localhost:3000/) (запуск может занять несколько секунд):

$ yarn start

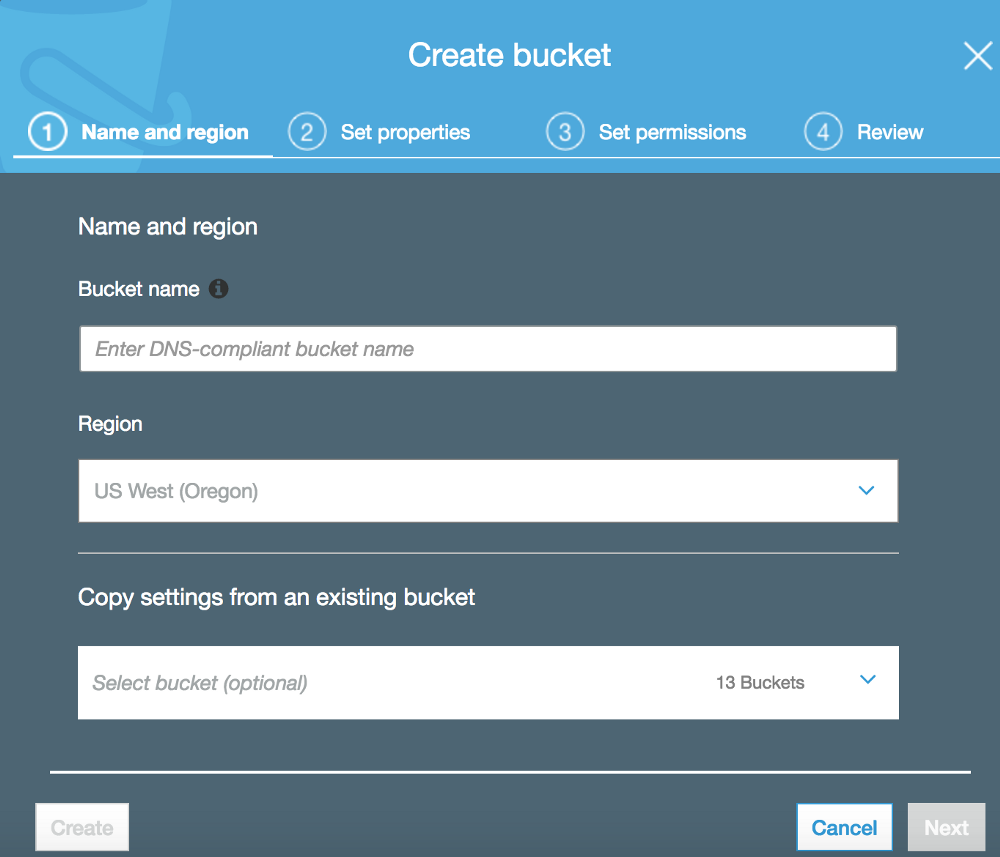
[](https://cdn.tproger.ru/wp-content/uploads/2017/07/1-8.png)

2. Настройка корзины AWS S3

Входим в свой аккаунт на AWS и выбираем S3:

[](https://cdn.tproger.ru/wp-content/uploads/2017/07/2-4.png)

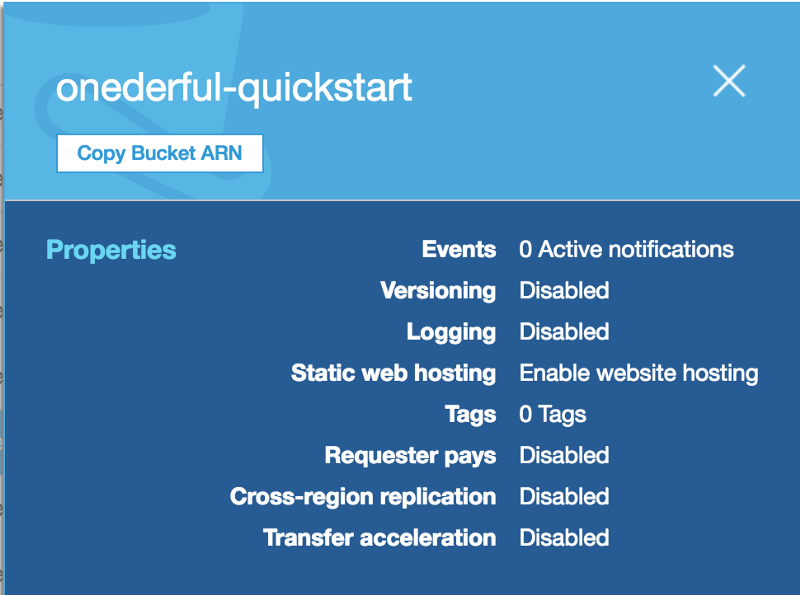
Нажимаем «Create bucket» и вводим имя (например: onederful-quickstart). Нажимаем «Далее» на всех остальных шагах и создаем [корзину](https://aws.amazon.com/ru/s3/details/) (bucket, в русскоязычных источниках также можно встретить термин «бакет»):

[](https://cdn.tproger.ru/wp-content/uploads/2017/07/3-2.png)

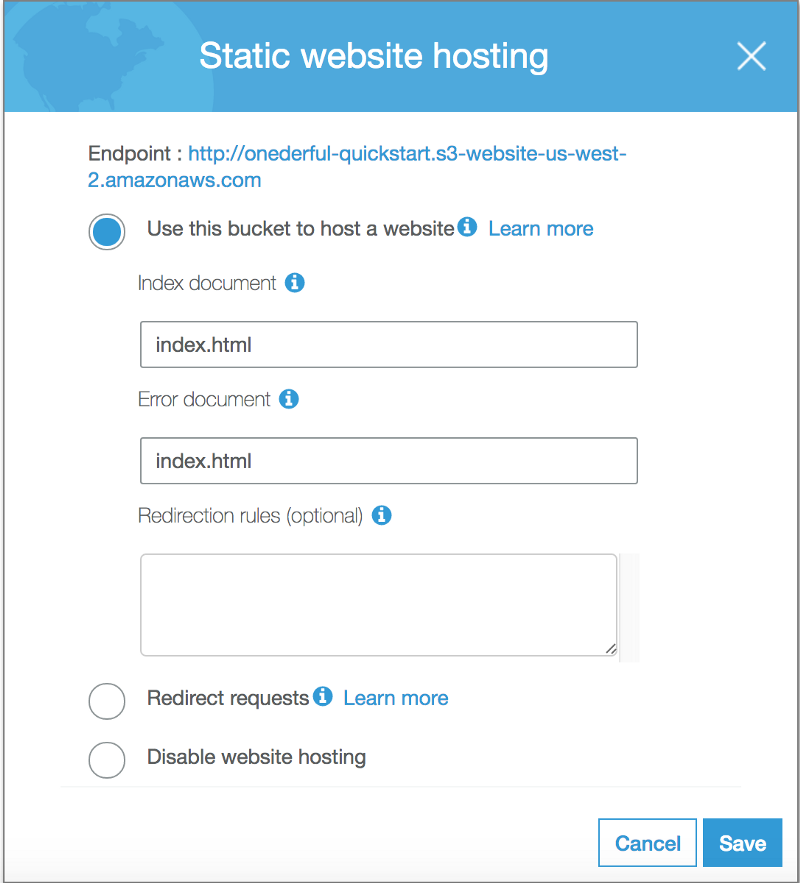
Теперь открываем только что созданную корзину:

[https://cdn.tproger.ru/wp-content/uploads/2017/07/4-2.png](https://cdn.tproger.ru/wp-content/uploads/2017/07/4-2.png)

После появления всплывающего окна нажимаем на «Properties»:

[](https://cdn.tproger.ru/wp-content/uploads/2017/07/5-2.png)

Нажимаем на «Static website hosting» и вводим «index.html» в каждом из полей «Index document» и «Error document». Теперь у нас есть общедоступный URL:

[](https://cdn.tproger.ru/wp-content/uploads/2017/07/6-3.png)

Переходим на вкладку «Permissions», вместо [YOUR BUCKET NAME] вписываем свое название проекта:

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "AllowPublicRead",

"Effect": "Allow",

"Principal": "\*",

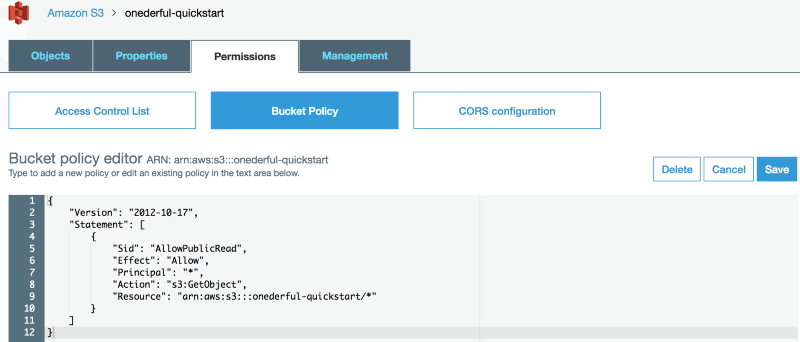
"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::[YOUR BUCKET NAME]/\*"

}

]

}

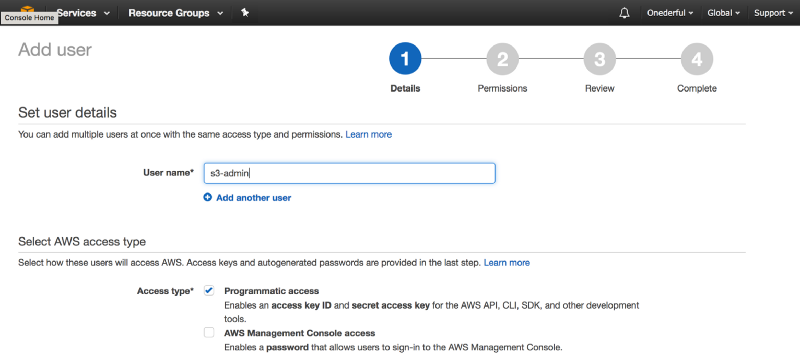
[](https://cdn.tproger.ru/wp-content/uploads/2017/07/7-2.png)

3. Создаем учетные данные пользователя AWS для загрузки файлов на S3

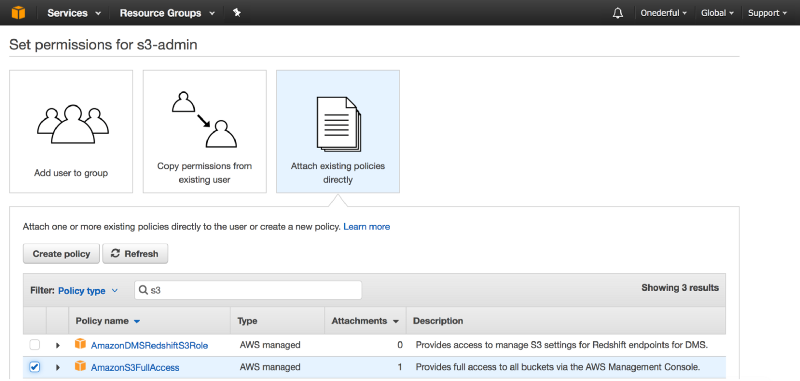
В консоли управления AWS нажимаем на «IAM» (Identity Access Manager):

[](https://cdn.tproger.ru/wp-content/uploads/2017/07/8-3.png)

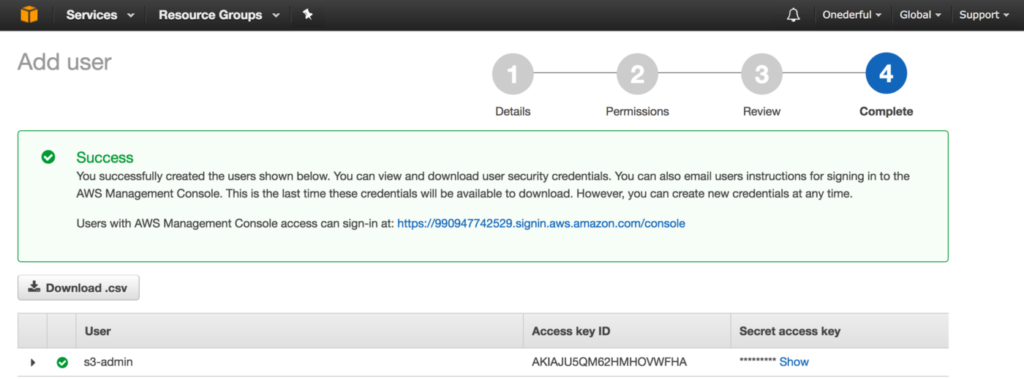
Переходим на вкладку «Users», находящуюся на боковой панели, и добавляем пользователя с именем «s3-admin»:

[](https://cdn.tproger.ru/wp-content/uploads/2017/07/9-2.png)

Прикрепляем «AmazonS3FullAccess policy»:

[](https://cdn.tproger.ru/wp-content/uploads/2017/07/10-2.png)

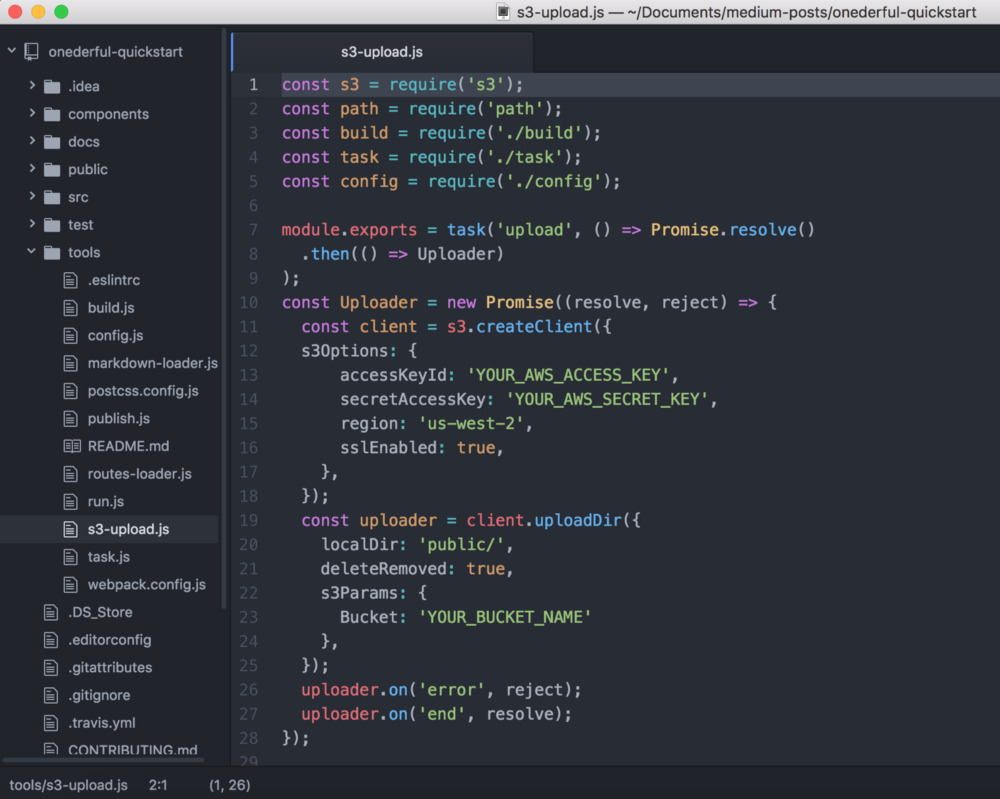
После создания пользователя сохраняем идентификатор доступа и секретный ключ (например, в блокноте) — они будут использоваться на последнем этапе этого руководства:

[](https://cdn.tproger.ru/wp-content/uploads/2017/07/11-2.png)

4. Публикуем шаблон на AWS

Замените следующие данные в файле tools/s3-upload.js:

* YOUR\_BUCKET\_NAME — на название корзины (со второго шага);
* YOUR\_AWS\_ACCESS\_KEY — на свой идентификатор доступа (с третьего шага);
* YOUR\_AWS\_SECRET\_KEY — на свой секретный ключ (с третьего шага).

[](https://cdn.tproger.ru/wp-content/uploads/2017/07/12-2.png)

Публикуем приложение:

$ yarn publish:webapp

5. Проверяем работоспособность и начинаем создавать приложение

Проверьте работоспособность приложения в вашем браузере. Если все работает, вы можете приступать к созданию логики вашего веб-приложения.

Настройка AWS должна выполняться только один раз, поэтому после внесения каких-либо изменений в ваше приложение можно просто запустить команду deploy, и в течение нескольких секунд изменения вступят в силу.