



11 лучших бессерверных фреймворков в 2024 году

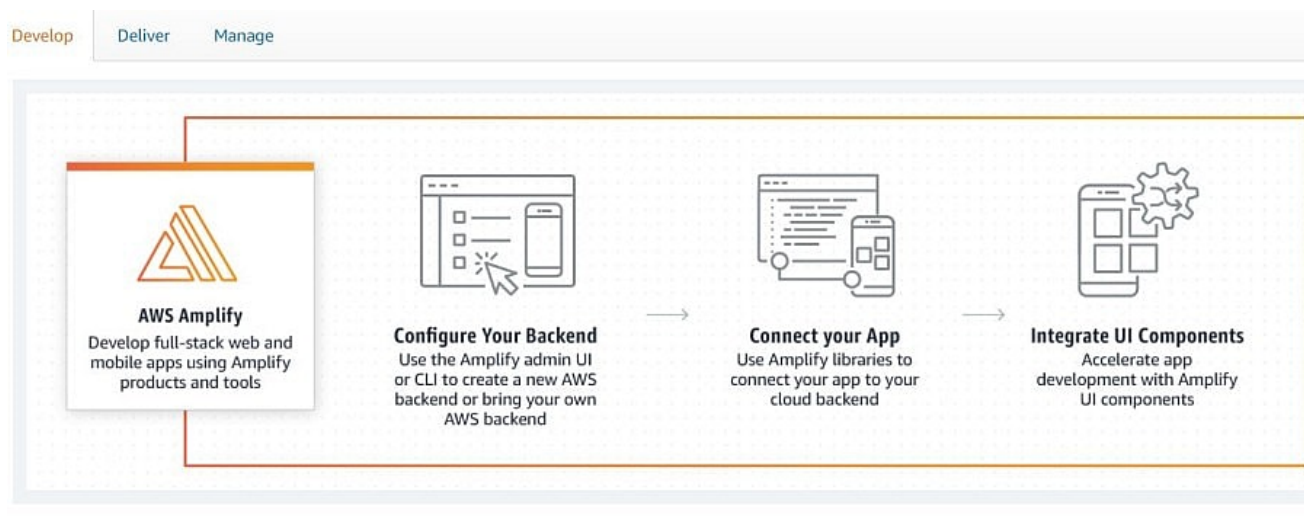
Описание

Вы увлечены технологией Serverless? Вам не терпится заняться новым проектом, но вы не уверены, с чего начать? Представляем вам обзор самых популярных фреймворков Serverless, которые вы можете взять в руки! Бессерверные вычисления – это новая тенденция для тех, кому нравится идея, что управлять нужно меньше, а времени на создание больше. И хотя эта технология все еще довольно новая, многие разработчики и инженеры прилагают все усилия для создания интересных решений для будущего бессерверных операций. Уверен, вы уже знаете, что Serverless не означает "без сервера". Это просто означает **меньше серверов и больше функций**. И если вы ищете надежную платформу FaaS (Functions as a Service) для своего следующего проекта, ознакомьтесь с нашей предыдущей статьей о лучших платформах для бессерверных вычислений. Итак, давайте рассмотрим доступные нам фреймворки. В завершение мы расскажем о некоторых интересных проектах, разрабатываемых в области Serverless, так что обязательно оставайтесь с нами!

AWS Amplify

Быстро создавайте масштабируемые бессерверные приложения для веб и мобильных устройств с помощью фреймворка AWS Amplify и получайте бесконечную гибкость. Настраивайте бессерверные бэкенды с данными, хранилищами, аутентификацией и многим другим с помощью интеллектуальных рабочих процессов. Подключайте мобильные и веб-приложения к существующим или новым ресурсам AWS с помощью меньшего количества строк кода. Вы можете размещать и

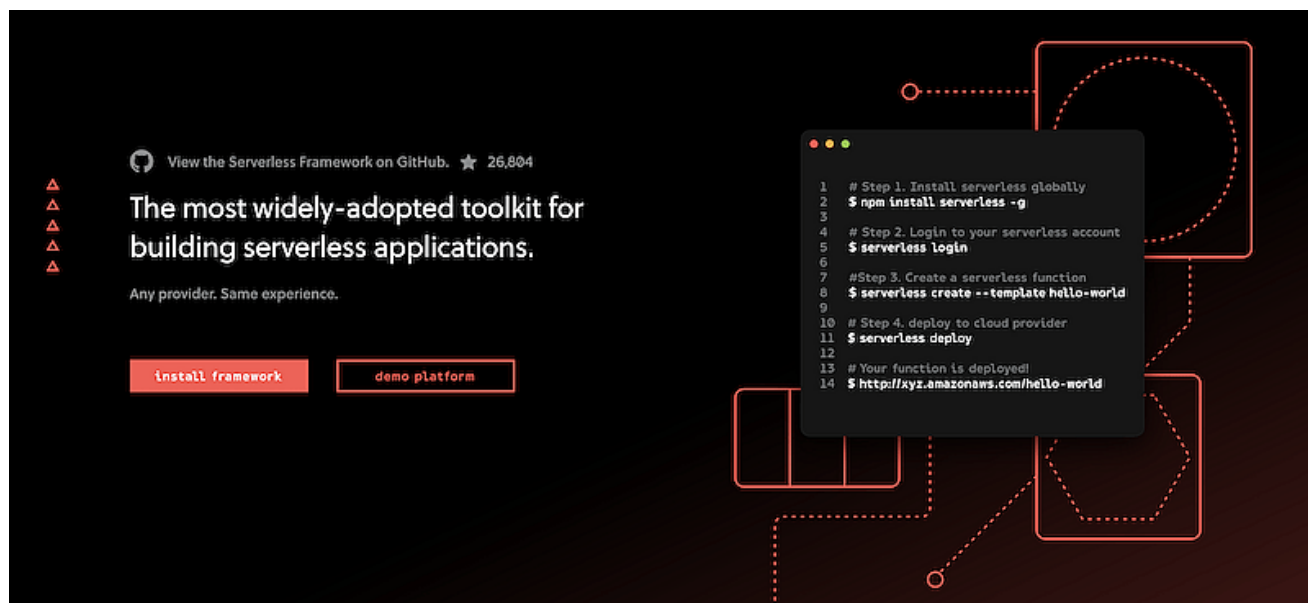
развертывать одностраничные веб-приложения, приложения с рендерингом на стороне сервера и статусные веб-сайты всего за несколько кликов. Кроме того, с его помощью можно легко управлять контентом и пользователями приложений.



AWS Amplify помогает внешним мобильным и веб-разработчикам быстрее создавать полнофункциональные приложения на AWS и настраивать их, когда вам это необходимо. Amplify поддерживает множество веб-фреймворков, таких как React, Angular, JavaScript, Vue.js, Next.js и т. д., и мобильных платформ, таких как iOS, Ionic, Flutter, Android и React Native. Вы можете интегрироваться со многими компонентами пользовательского интерфейса Amplify, чтобы ускорить разработку приложений. Получите множество функций и инструментов, таких как аутентификация, взаимодействия, функции, DataStore, управление пользователями, API (REST, GraphQL), аналитика, AI/ML-прогнозы, хранение, PubSub, push-уведомления и т. д., для разработки. Кроме того, к вашим услугам управляемый хостинг, предварительный просмотр PR, CI/CD, пользовательские домены и мониторинг развертывания.

Бессерверные

Я знаю, что вы думаете: “Кому-то повезло получить доменное имя Serverless.com, и он просто наживается на своей удаче!”, и я должен сказать вам, что это не так.



Serverless Framework имеет около 30 000 звезд на GitHub, что делает его самым популярным бессерверным фреймворком, который вы можете найти. Итак, что вы можете сделать с помощью этого фреймворка? Прежде всего, вы можете создавать приложения, использующие микросервисы как средство реагирования на события и т. д. Например, ваши приложения будут реагировать только при наличии определенного вызова и масштабироваться в зависимости от потребности. В результате вы можете сэкономить кучу денег, которые в противном случае были бы потрачены на поддержание неработающих сервисов. Framework использует новые сервисы вычислений, управляемые событиями, такие как AWS Lambda, Google Cloud Functions и другие. Это инструмент командной строки, предоставляющий леса, автоматизацию рабочих процессов и лучшие практики для разработки и развертывания бессерверной архитектуры. Он также полностью расширяем с помощью плагинов.

Midway

Получите бессерверный фреймворк Node.js для создания традиционных серверных приложений, небольших программ, микросервисов и построения бессерверных сервисов с помощью Midway. Он написан на языке typescript и использует механизм инъекций IoC для разделения логики приложения и более естественной разработки больших приложений на Node.js. В качестве базовых фреймворков Midway может использовать express, EggJS или koa. Он предоставляет такие базовые решения, как GRPC, Dubbo.js, RabbitMQ и Socket.io для самостоятельного использования. Кроме

того, Midway работает как бессерверный веб-фреймворк Node.js для full-stack или front-end разработчиков.

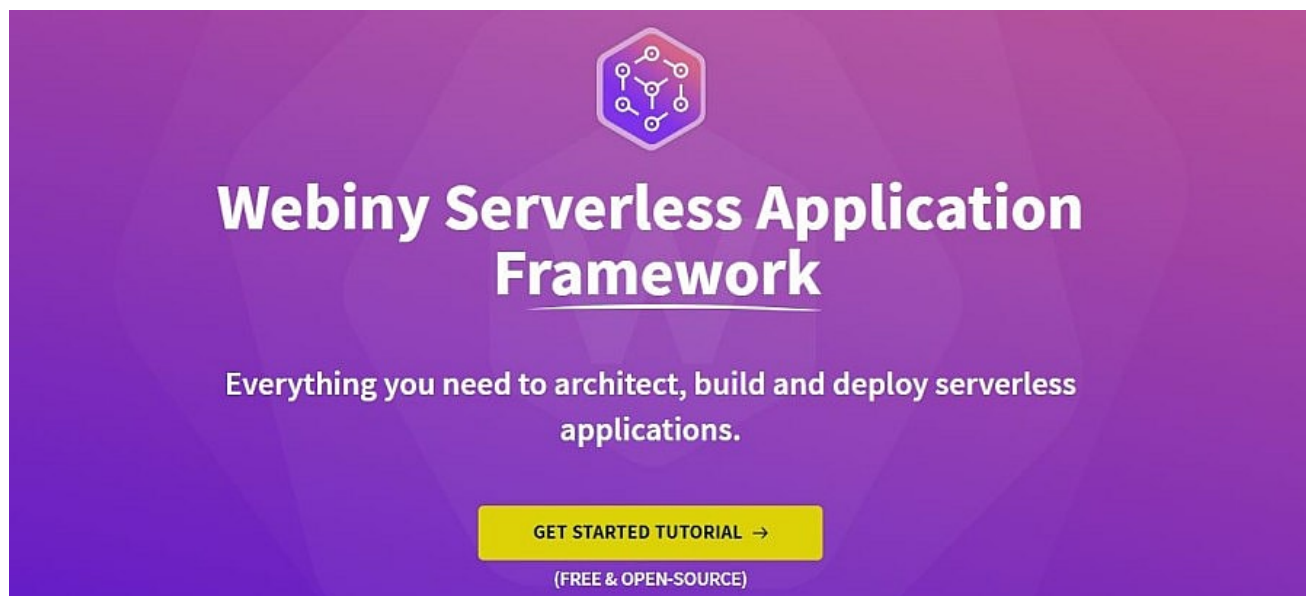


Midway позволяет создавать приложения нового поколения, работающие в контейнерах, на виртуальных машинах, в Alibaba Cloud, AWS и Tencent Cloud. Вы также можете легко интегрировать его с Vue и React. Вы найдете множество функций, таких как:

- Групповые и масштабируемые плагины
- Совместимость и архитектура промежуточного ПО продвинутой группы
- Разделение возможностей и слоев приложений
- Опыт разработки нового поколения
- Поддерживает промежуточное ПО коа и плагины Egg.

Webiny

Получите все необходимое для проектирования, развертывания и создания бессерверных приложений с помощью фреймворка Webiny Serverless. Он предназначен для различных сценариев использования. Вы можете изменять архитектуру, технологический стек и адаптироваться к потребностям вашей организации и проекта. Создавайте бизнес-сайты и приложения на основе SEO, маршрутизации, предварительного рендеринга, безопасности, развертывания, комплектации и многого другого. Создавайте внутренние и внешние API, извлекая резолверы GraphQL. Не ограничивайтесь одним приложением – создавайте приложения, использующие различные базы данных. Вы можете использовать драйверы баз данных NodeJS для взаимодействия с внешними базами данных.



Webiny обеспечивает пакетирование кода, IaC, механизмы развертывания и организацию проектов. Добавьте продукт в топ-лист и предложите SaaS всему миру. Защитите часть или все приложения с помощью ACL с помощью встроенного AWS Cognito и фреймворка безопасности. Добавьте поддержку Auth0, OKTA и других. Вы получите безголовую CMS, конструктор страниц, конструктор форм, файловый менеджер и многое другое. Генерируйте шаблонный код с помощью CLI-плагинов, начиная с лямбда-функций и заканчивая новыми GraphQL API. Пользовательский интерфейс администратора Webiny содержит более тридцати различных компонентов React. Вы также получите защиту от DDoS, CDN и долговечность данных 11 9. Webiny экономит ваше время на пробы, ошибки и исследования.

AWS Chalice

Вы ищете фреймворк для написания бессерверных приложений на Python? Попробуйте AWS Chalice, который позволяет создавать и развертывать приложения через AWS Lambda. Chalice предоставляет инструмент командной строки для создания, развертывания и управления приложениями, API на основе декораторов для интеграции с Amazon S3, Amazon API Gateway, Amazon SQS, Amazon SNS и другими сервисами AWS.

CHALICE

Он позволяет создавать REST API, выполнять периодически запускаемые задачи, связывать лямбда-функции с событиями S3 и очередью SQS, а также использовать другие ресурсы AWS. После написания кода Chalice позволит вам быстро запустить и развернуть ваше приложение. Chalice поддерживает все версии Python, которые поддерживает AWS Lambda, включая Python 2.7, 3.6, 3.7 и 3.8. Настройте все учетные данные перед развертыванием приложения. Но если вы уже настроили его для работы с AWS CLI или boto3, пропустите этот пункт. После развертывания вы можете внести изменения там, где вам нужно, и повторно развернуть изменения. Изучайте учебники, темы и ссылки на API, чтобы улучшить свои знания.

Zappa

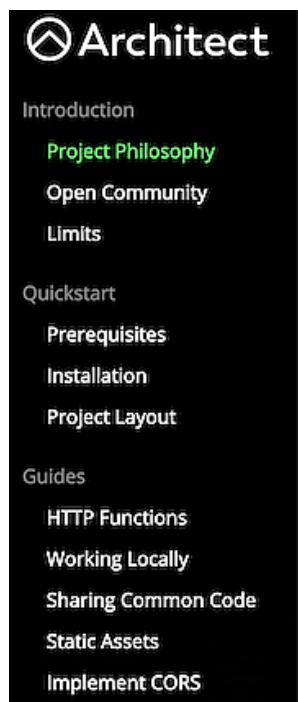
Zappa – это бессерверный фреймворк для создания приложений на Python. Он облегчает работу по созданию и развертыванию бессерверных и событийно-управляемых приложений Python на API Gateway и AWS Lambda. Используйте его, чтобы насладиться нулевым обслуживанием, нулевым временем простоя и бесконечным масштабированием при минимальных затратах на текущие развертывания. Прежде чем приступить к установке и настройке, убедитесь, что у вас есть Python 3.x, учетная запись AWS и правильная установка файла учетных данных AWS. Вы также можете легко установить Zappa с помощью pip.



Zappa автоматически настраивает параметры развертывания. После того как настройки будут правильно настроены, разверните приложение на этапе производства с помощью одной командной строки. Используйте команду `template`, если вам нужен шаблон API Gateway CloudFormation. Разверните Zappa на пользовательских поддоменах и доменных именах с помощью SSL-сертификатов, сертификатов AWS Certificate Manager и сертификатов Let's Encrypt. Включите безопасные конечные точки на шлюзе API, такие как ключ API, политика IAM, авторизаторы лямбда-шлюза API, авторизатор пула пользователей Cognito и политика ресурсов шлюза API. Кроме того, вы можете установить переменные окружения: локальное окружение, удаленное окружение AWS и удаленное окружение через файл S3,

Архитектор

Создавайте, развертывайте и поддерживайте бессерверную инфраструктуру нового поколения на основе облачных функций AWS с полной локализацией, автономными рабочими процессами и многим другим.



```
@tables
likes
  likeID *String
  update Lambda
```

```
@indexes
likes
  date *String
```

Running `npm create` in the same directory as the `.arc` file above generates the following function code:

```
/
├─ src
│   └─ http
│       ├── get-index/
│       ├── get-likes/
│       └─ post-likes/
│   └─ events
│       └─ hit-counter/
│   └─ scheduled
│       └─ daily-affirmation/
│   └─ tables
│       ├── likes-update/
│       └─ shared/
```

Это достаточно сложный проект, и вам необходимо ознакомиться с философией проекта и страницей документации в целом.

Claudia.js

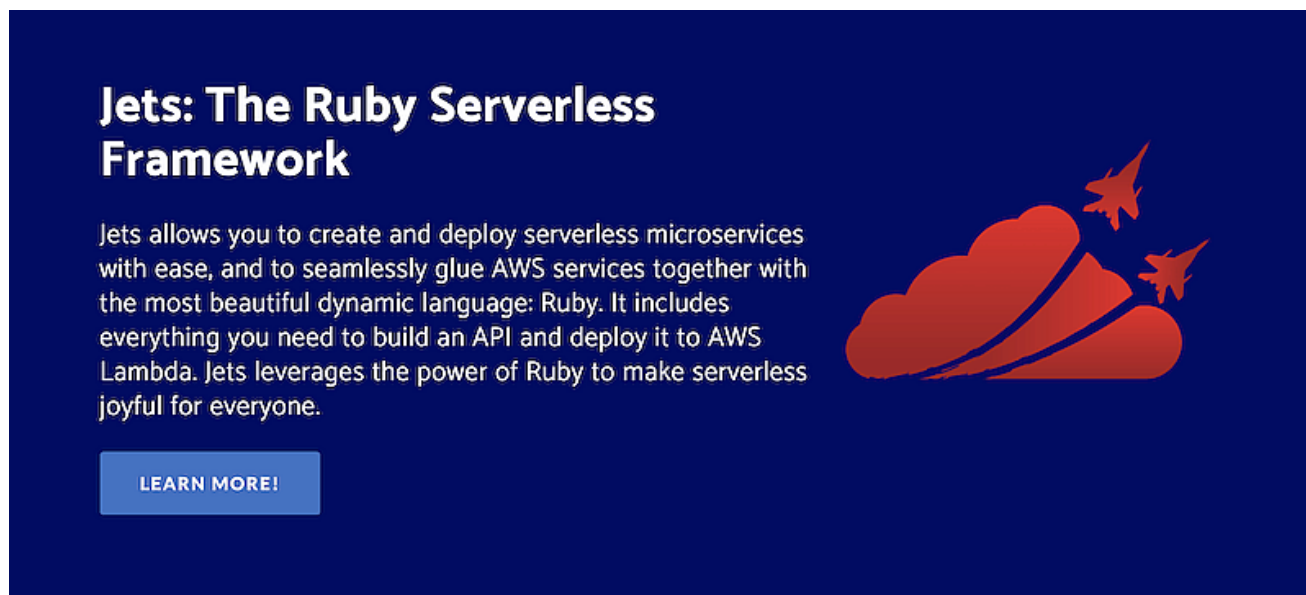
Claudia.js предоставляет разработчикам средства для развертывания проектов на базе Node.js в системах Lambda и API. Claudia автоматизирует задачи развертывания, конфигурирования и автоматически настроит вашу среду JavaScript для современной разработки. Итак, что это значит для вас? Используя Claudia, вы сможете начать свой путь к микросервисам за считанные минуты. В результате у вас останется гораздо меньше времени на раздумья и гораздо больше времени, чтобы сосредоточиться на решении реальных проблем. Основные причины использования Claudia включают в себя:

- Развертывание, обновление и управление экземплярами с помощью одной команды
- Безупречно работает с пакетами NPM
- Интегрированная структура шаблонов позволит вам сосредоточиться на выполнении поставленных задач.
- Управление версиями, позволяющее работать над несколькими версиями одновременно

- Небольшая кривая обучения

Jets

Вы любите Руби?



Jets: The Ruby Serverless Framework

Jets allows you to create and deploy serverless microservices with ease, and to seamlessly glue AWS services together with the most beautiful dynamic language: Ruby. It includes everything you need to build an API and deploy it to AWS Lambda. Jets leverages the power of Ruby to make serverless joyful for everyone.

[LEARN MORE!](#)

Тогда вам может понравиться Jets – бессерверный Ruby-фреймворк для создания и развертывания микросервисов. Фреймворк включает в себя все необходимые инструменты для создания API-структур и их развертывания на Lambda. Jets также подходит для написания отдельных функций, которые можно использовать для объединения различных сервисов и ресурсов, предоставляемых AWS. Принцип работы Jets заключается в том, что фреймворк преобразует ваш существующий код в функции для Lambda и других сервисов AWS. В результате вы можете пропустить такие вещи, как развертывание и инициализация, и потратить больше времени на написание функционального кода.

Pulumi

Pulumi предоставляет бесшовный SDK, который поможет вам создавать и развертывать программное обеспечение на любой облачной платформе. SDK поддерживает контейнеры, размещенные сервисы, основную инфраструктуру и контейнеры.



Cloud Native Infrastructure as Code

Build software for any cloud using your favorite language.

Вы можете писать код на любом языке, а Pulumi будет предоставлять и управлять вашими ресурсами на таких провайдерах, как GCP, AWS, Kubernetes и Azure.

Примеры бессерверных технологий

Мы уже рассказывали о Serverless Framework ранее в этом посте, и, по сути, это ресурс, демонстрирующий множество различных вариантов использования Serverless Framework.

Java AWS Simple HTTP Endpoint This example demonstrates how to setup a simple HTTP GET endpoint using Java. Once you ping it, it will reply with the current time. go to github >	nodeJS AWS Alexa Skill Example This example demonstrates how to setup your own Alexa skill using AWS Lambdas. go to github >	nodeJS AWS Auth0 Cognito Custom Authorizers API Authorize your API Gateway with either Auth0 or Cognito JWKS RS256 tokens. go to github >	nodeJS AWS Auth0 Custom Authorizers API This is an example of how to protect API endpoints with Auth0, JSON Web Tokens (jwt) and a custom authorizer lambda function. go to github >
nodeJS AWS DynamoDB Backup Example This examples shows your how to create a backup of your DynamoDB table to S3. go to github >	nodeJS AWS Env Variables Encrypted In A File This example demonstrates how to store secrets like API keys encrypted in your repository while providing them as environment variables to your AWS Lambda functions. go to github >	nodeJS AWS Env Variables Example This example demonstrates how to use environment variables for AWS Lambdas. go to github >	nodeJS AWS Fetch File And Store In S3 This example display how to fetch an image from remote source (URL) and then upload this image to a S3 bucket. go to github >

Несмотря на то что эти примеры привязаны к одному конкретному фреймворку, вы все равно сможете насладиться процессом знакомства с новыми концепциями

serverless.

Дата Создания

29.03.2024