

УДК 004.05

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПЛАТФОРМ ДЛЯ РАЗРАБОТКИ И РАЗВЕРТЫВАНИЯ БЕССЕРВЕРНЫХ ПРИЛОЖЕНИЙ В ЭКОСИСТЕМЕ PYTHON

**Мишина Владлена Игоревна,**

студент университета ИТМО, Санкт-Петербург, электронная почта: vladaa905@gmail.com

### Аннотация

В статье рассмотрены критерии, позволяющие сравнивать платформы, используемые при разработке бессерверных веб-приложений, а также проведено сравнительное исследование основных платформ для разработки бессерверных веб-приложений (AWS, Google Cloud Functions, Microsoft Azure Cloud Functions, Yandex Cloud Functions) с точки зрения разработки в экосистеме Python, выявлены их схожие характеристики и различия, определены факторы, на которые следует обращать внимание при выборе облачного провайдера.

**Ключевые слова:** serverless, веб-приложение, Python, функции, облачные сервисы, AWS, Google Cloud Functions, Microsoft Azure Cloud Functions, Yandex Cloud Functions.

## COMPARATIVE ANALYSIS OF PLATFORMS FOR DEVELOPING AND DEPLOYING SERVERLESS APPLICATIONS IN THE PYTHON ECOSYSTEM

**Vladlena I. Mishina,**

student of ITMO University, St. Petersburg, e-mail: vladaa905@gmail.com

### ABSTRACT

The article discusses criteria that allow to compare platforms used in the development of serverless web applications, as well as a comparative study of the main platforms for developing serverless web applications (AWS, Google Cloud Functions, Microsoft Azure Cloud Functions, Yandex Cloud Functions) from a development point of view in the Python ecosystem, identified their similar characteristics and differences, factors that should be paid attention to when choosing a cloud provider.

**Keywords:** serverless, web application, Python, functions, cloud services, AWS, Google Cloud Functions, Microsoft Azure Cloud Functions, Yandex Cloud Functions.

Serverless — сравнительно новая, но активно набирающая популярность технология. Она дает разработчикам возможность сосредоточиться полностью только на реализации

бизнес-логики и практически не думать об инфраструктурных деталях. В частности, тенденция применения бессерверных подходов растет и в России. Согласно исследованию, проведенному Yandex Cloud и исследовательской компанией Ipsos в декабре 2024 года, 83% участвовавших представителей IT-индустрии согласились, что количество проектов, использующих serverless-технологии, будет только расти в 2024 году [1]. Актуальность работы обусловлена тем, что ключевой особенностью serverless-технологии является делегирование задач по управлению инфраструктурой, таких как обслуживание, масштабирование серверов, балансировка нагрузки между ними поставщику облачных услуг. В этой связи выбор такого поставщика имеет чрезвычайно важное значение.

Целью исследования является проведение сравнительного анализа платформ для разработки и развертывания бессерверных приложений, в частности, в экосистеме одного из наиболее распространенных языков программирования Python.

Поскольку Python является одним из лидирующих языков программирования, он поддерживается большинством бессерверных платформ. Однако в зависимости от особенностей каждой платформы скорость и удобство разработки может значительно отличаться. Для выявления таких особенностей и сравнения платформ определены следующие критерии.

1. Версии Python, поддерживаемые платформой
2. Наличие и полнота документация

Является одним из основополагающих требований к любому программному продукту. Важно оценить, насколько полно документация описывает все функции платформы, информирует о том, как разработчик могут с ней работать. В частности, необходимо оценить, насколько она освещает моменты, связанные с использованием Python при разработке приложений.

3. Поддержка инструментов управление версиями

Важным аспектом платформ, который может упростить несколько отдельных этапов жизненного цикла приложения, является управление версиями бессерверных приложений и функций. Помимо облегчения процесса разработки, управление версиями также полезно для развертывания приложений. Наиболее благоприятно, если платформа предоставляет механизмы для управления версиями как на уровне отдельных версий функций, так и целых версий приложений, т. е. комбинации функций и ресурсов, отслеживаемых вместе [2].

4. Интерфейс

Существующие платформы предлагают различные способы взаимодействия с ними, и цель критерия «интерфейс» состоит в том, чтобы классифицировать платформу по этому параметру. Данный критерий позволяет перечислить поддерживаемые типы интерфейсов, т. е. предлагают ли они интерфейс командной строки (CLI), интерфейс прикладного программирования (API) и/или графический интерфейс пользователя (GUI) [2].

5. Размер пакета развертывания
6. Хранение данных
7. Источники триггеров

В данный критерий входят, например, события, которые могут запустить выполнение бессерверных функций.

8. Максимальное количество функций
9. Максимальный объем памяти, который может использовать каждая функция

## 10. Время жизни функции

Это время, в течение которого функция остается активной и реагирует на обработку запросов. Этот показатель напрямую влияет на затраты и производительность, а понимание этого срока помогает оптимизировать использование ресурсов, минимизировать задержки и повысит эффективность управления расходами.

## 11. Поддерживаемые платформой фреймворки

Современная разработка веб-приложений практически не представляется возможной без использования фреймворков, которые не только упрощают процесс написания приложения, но и повышают скорость и эффективность. В связи с этим, важно определить какие совместимые с Python бессерверные фреймворки поддерживает платформа.

## 12. Тестирование, мониторинг и отладка

Важный набор механизмов связан с тестированием и отладкой автономных функций и бессерверных приложений. Хотя платформы обычно не несут ответственности за разработку кода, могут быть предоставлены дополнительные способы тестирования функциональных и нефункциональных аспектов развернутых функций, например модульное и интеграционное тестирование или нагрузочное тестирование. Например, платформа может предлагать механизмы для облегчения модульного тестирования с помощью специфичных для платформы библиотек или проверки вызова функций с помощью специальных команд CLI. Кроме того, платформа может предоставлять комбинацию локальных и удаленных механизмов отладки [2].

В результате анализа зарубежных и отечественных источников, посвященных исследованиям в области разработки serverless веб-приложений, можно выделить несколько наиболее популярных решений (AWS, Google Cloud Functions, Microsoft Azure Cloud Functions, Yandex Cloud Functions), которые будут рассмотрены далее.

В таблице 1 представлено сравнение наиболее широко используемых платформ, в частности с одной из самых популярных в России отечественных платформ.

Таблица 1 – Сравнительная таблица платформ для разработки и развертывания бессерверных приложений

Критерий	Amazon Web Service	Microsoft Azure	Google Cloud Platform	Yandex.Cloud
Версии Python, поддерживаемые платформой	Версии до Python 3.12	Версии до Python 3.11	Версии до Python 3.12	Версии до Python 3.12
Наличие и полнота документации	Имеется все необходимая документация для разработки и развертывания бессерверных приложений на платформе, включая подробные	Имеется все необходимая документация для разработки и развертывания бессерверных приложений на платформе, включая	Имеется все необходимая документация для разработки и развертывания бессерверных приложений на платформе, включает	Имеется все необходимая документация для разработки и развертывания бессерверных приложений на платформе, включает

	руководства, учебные пособия и справочники по API для своих сервисов. Регулярно обновляется	краткие руководства, учебные пособия и справочные материалы. Интегрирует свою документацию с более широкой экосистемой Microsoft.	краткие руководства, практические руководства и лучшие практики. Ориентирована на реальные приложения и интеграцию с другими сервисами и API Google.	краткие руководства, практические руководства.
Поддержка инструментов управления версиями	Управление версиями на уровне функций и приложений с помощью шаблонов AWS SAM.	Нет отдельного инструмента	Нет отдельного инструмента	Нет отдельного инструмента
Интерфейс	Интерфейс командной строки (CLI), интерфейс прикладного программирования (API), графический интерфейс пользователя (GUI)	Интерфейс командной строки (CLI), интерфейс прикладного программирования (API), графический интерфейс пользователя	Интерфейс командной строки (CLI), интерфейс прикладного программирования (API), графический интерфейс пользователя	Интерфейс командной строки (CLI), интерфейс прикладного программирования (API), графический интерфейс пользователя (GUI)
Размер пакета развертывания	Не более 50 МБ для архивированных файлов и 250 МБ для не архивированных	Не более 1 ГБ	Не более 100 МБ для архивированных файлов и 500 МБ для не архивированных	ZIP-архив через консоль управления - 3,5 МБ, ZIP-архив через S3 бакет - 128 МБ, распакованный ZIP-архив через S3 бакет - 680 МБ
Хранение данных	Amazon DynamoDB, Amazon SimpleDB and S3	Azure Cosmos DB and Azure Storage	Cloud Storage, Cloud Firestore and Cloud Datastore	YDB
Источники триггеров	AWS API Gateway, Amazon	Azure Cosmos DB, Azure Event	Google cloud API Gateway,	Native http triggers, Yandex API Gateway

	DynamoDB, Amazon Kinesis Data Streams, Amazon S3, API Gateway, Amazon SNS, and Amazon CloudWatch	Hubs, Azure Event Grid, Azure Mobile Apps, Azure Notification Hubs, Azure Service Bus, Azure Storage, GitHub, HTTP, триггеры по времени.	Azure Grid, Cloud Pub/Sub, Cloud Storage and HTTP requests	
Максимальное количество функций	Не ограничено	Не ограничено	1000 функций на проект	10 функций на одно облако
Максимальный объем памяти, который может использовать каждая функция	10 240 МБ	1,5 ГБ	8 192 МБ	20 ГБ для всех функций
Время жизни функции	До 15 мин.	До 10 мин.	До 10 мин.	До 10 мин.
Поддерживаемые фреймворки	Serverless framework, Zappa, Chalice	Serverless framework	Serverless framework	Yappa
Тестирование, мониторинг и отладка	CloudWatch and X-Ray, AWS SAM для тестирования [3]	Application Insights[4]	Stackdriver Monitoring [5]	Сервис Monitoring [6]

В результате сравнения можно сделать выводы о том, что все платформы поддерживают актуальные версии Python, имеют достаточно обширную документацию и несколько интерфейсов, а также имеют приблизительно одинаковое время жизни serverless-функций. Исходя из максимального количества функций и памяти, которые они могут использовать можно отметить, что для больших масштабных проектов оптимальными будут платформы Amazon Web Service и Google Cloud Platform. Наиболее развитые инфраструктуры имеют зарубежные решения. В частности, Amazon Web Service оказалась единственной платформой, которая имеет документацию, описывающую управление версиями на уровне функций и приложений, где приложения можно определять и управлять ими с помощью AWS SAM шаблонов. Для сравнительно небольших проектов подходящим и доступным решением будет Yandex.Cloud, который также имеет достаточно развитую экосистему и обладает достаточным функционалом для бессерверной разработки и развертывания Python веб-приложений.

В заключение можно сделать вывод о том, что все рассмотренные решения обладают достаточным функционалом для разработки и развертывания веб-приложений с использованием Python. Выбор платформы помимо такого очевидного аспекта как ценовая политика зависит от характеристик разрабатываемого приложения, в частности, основное внимание следует уделять его предполагаемому размеру и потенциальному расширению, фреймворкам используемым при его разработке, а так же специфическим особенностям

экосистемы платформы. Продолжением исследования может стать более подробное изучение отечественных облачных платформ, их экосистем, возможностей, которые они предоставляют и особенностей их использования при разработке бессерверных веб-приложений на Python.

#### Список литературы:

1. The State of Serverless: исследование Yandex Cloud и Ipsos [Электронный ресурс].- 2024.- URL: [https://cloud.yandex.ru/ru/blog/posts/2023/12/the-state-of-serverless?utm\\_source=telegram&utm\\_medium=seed&utm\\_campaign=frontendnoteschannel&utm\\_content=results-of-research&utm\\_term=01\\_2024](https://cloud.yandex.ru/ru/blog/posts/2023/12/the-state-of-serverless?utm_source=telegram&utm_medium=seed&utm_campaign=frontendnoteschannel&utm_content=results-of-research&utm_term=01_2024)&(Дата обращения: 10.03.2024)
2. Yussupov V., Soldani J., Breitenbücher U., Brogi A., Leymann F. FaaS: your decisions: A classification framework and technology review of function-as-a-Service platforms // Journal of Systems and Software. - 2021. - V.175.
3. AWS Documentation [Электронный ресурс]. - 2024. - URL: [https://docs.aws.amazon.com/?nc2=h\\_ql\\_doc\\_do](https://docs.aws.amazon.com/?nc2=h_ql_doc_do) (Дата обращения: 10.03.2024)
4. Azure documentation [Электронный ресурс]. - 2024. - URL: <https://learn.microsoft.com/en-us/azure/?product=popular> (Дата обращения: 10.03.2024)
5. Google Cloud documentation [Электронный ресурс]. - 2024. - URL: <https://cloud.google.com/docs> (Дата обращения: 10.03.2024)
6. Документация Yandex.Cloud [Электронный ресурс]. - 2024. - URL: <https://cloud.yandex.ru/docs> (Дата обращения: 10.03.2024)

#### References:

1. The State of Serverless: a study by Yandex Cloud and Ipsos [Electronic resource].-2024.- URL: [https://cloud.yandex.ru/ru/blog/posts/2023/12/the-state-of-serverless?utm\\_source=telegram&utm\\_medium=seed&utm\\_campaign=frontendnoteschannel&utm\\_content=results-of-research&utm\\_term=01\\_2024](https://cloud.yandex.ru/ru/blog/posts/2023/12/the-state-of-serverless?utm_source=telegram&utm_medium=seed&utm_campaign=frontendnoteschannel&utm_content=results-of-research&utm_term=01_2024)&( Accessed: 10.03.2024)
2. Yussupov V., Soldani J., Breitenbücher U., Brogi A., Leymann F. FaaS: your decisions: A classification framework and technology review of function-as-a-Service platforms // Journal of Systems and Software. - 2021. - V.175.
3. AWS Documentation [Electronic resource]. - 2024. - URL: [https://docs.aws.amazon.com/?nc2=h\\_ql\\_doc\\_do](https://docs.aws.amazon.com/?nc2=h_ql_doc_do) (Accessed: 10.03.2024)
4. Azure documentation [Electronic resource]. - 2024. - URL: <https://learn.microsoft.com/en-us/azure/?product=popular> (Accessed: 10.03.2024)
5. Google Cloud documentation [Electronic resource]. - 2024. - URL: <https://cloud.google.com/docs> (Accessed: 10.03.2024)
6. Yandex.Cloud documentation [Electronic resource]. - 2024. - URL: <https://cloud.yandex.ru/docs> (Accessed: 10.03.2024)