

ОБЛАЧНАЯ РАЗРАБОТКА. ПРЕИМУЩЕСТВА И ОСОБЕННОСТИ

С.В. Зинченко, Ю.В. Балашова

Статья посвящена понятиям облака, облачных вычислений, облачной разработки и различием облачной разработки от стандартной модели разработки. Рассматриваются ключевые преимущества и разновидности облачной разработки программного обеспечения.

Ключевые слова: облако, облачные вычисления, облачная разработка, фреймворки, гибкость, конвейеры

CLOUD DEVELOPMENT. BENEFITS AND FEATURES

S.V. Zinchenko, Y.V. Balashova

The article is devoted to the concepts of cloud, cloud computing, cloud development and the difference between cloud development and the standard development model. The key advantages and varieties of cloud software development are considered.

Keywords: cloud, cloud computing, cloud development, frameworks, flexibility, pipelines

В последние годы в различные сферы деятельности общества проникает понятие «облако» (The cloud). Облако – это модель работы, при котором данные хранятся на распределенных серверах, доступ к которым осуществляется через интернет. Помимо хранения данных на серверах может быть исполняемое программное обеспечение и базы данных. Облачные сервера могут быть расположены географически в разных частях мира, но для пользователя являться

единым целым. Используя облачные технологии, пользователям и компаниям нет необходимости содержать и управлять физически своими серверами или запускать приложения на своих компьютерных мощностях. Типичная организация облака показана на рисунке 1.

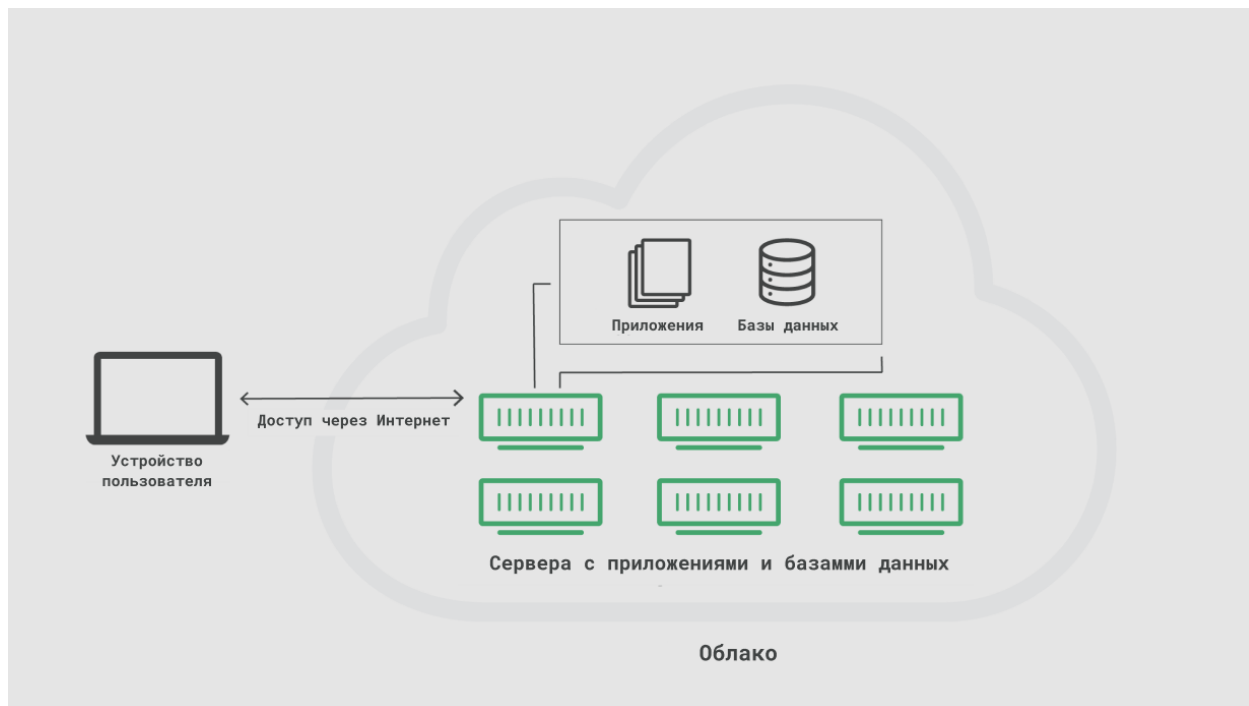


Рисунок 1 - Организация облака

Облачные технологии предоставляют ряд преимуществ:

1. **Затраты.** Используя облачные вычисления, можно исключить затраты на покупку и содержание серверов, специализированного программного обеспечения. Также исключаются затраты на содержание серверов и инфраструктуры, такие как электроснабжение, охлаждение и специализированный персонал для обслуживания.
2. **Гибкость.** За считанные минуты можно увеличивать и уменьшать выделенные вычислительные ресурсы.
3. **Производительность.** Облачные сервисы как правило регулярно обновляют аппаратную составляющую своих серверов, что позволяет пользователю использовать актуальные в данный момент аппаратные технологии и мощности

4. **Надежность.** Резервное копирование, зеркалирование, обеспечение непрерывности работы экономически выгоднее чем при содержании собственных серверов.

5. **Безопасность.** Многие поставщики облачных услуг предоставляют широкий спектр технологий безопасности и шифрования, помогающих в защите данных.

Существует три типа архитектуры облачных вычислений:

1. **Публичное облако (Public cloud).** Такие облака могут использоваться несколькими организациями одновременно, т.е. используются виртуальные машины и отдельные сервера используются разными пользователями. Это называется мультиарендность, поскольку несколько арендаторов арендуют пространство на одном сервере.

2. **Частное облако (Private cloud).** В этом случае это сервер, дата-центр или распределенная сеть, полностью выделенная одной организации.

3. **Гибридное облако (Hybrid cloud).** Гибридные облака могут объединять публичные и частные облака, а также локальные сервера организации. Например, можно использовать частное облако для одного вида услуг, а общедоступное для других целей.

Помимо вышеперечисленного, облачные технологии находят все более широкое применение в области разработки программного обеспечения. Облачная разработка (Cloud development) – это создание программного обеспечения, предназначенного для работы внутри облачной среды вычислений. Традиционный подход к разработке программного обеспечения как правило подразумевает разработку, тестирование и запуск приложений на компьютере или физических серверах. В тоже время при облачной разработке программное обеспечение разрабатывается, тестируется и запускается в облачной среде.

В течении последних нескольких лет произошли ряд событий, которые привели к запоздалому внедрению облачной разработки. При этом облачная разработка, существующая уже около 10 лет, только недавно начала становится стандартом разработки. На это повлияли следующие тенденции:

1. Распространение **Software-as-a-Service (SaaS)** (программное обеспечение как услуга) продуктов, являющихся по сути облачными программными приложениями.

2. Рост количества организаций, запускающих программное обеспечение в облачной среде.

3. Использование контейнерных технологий, таких как **Kubernetes** и **Docker**. В этом случае программное обеспечение не зависит от среды его выполнения.

4. Появление более сложного программного обеспечения и использование **AI/Machine Learning** (искусственный интеллект/машинное обучение) требует больших вычислительных мощностей, которые как правило превосходят возможности обычных персональных компьютеров.

5. В последние годы стоимость облаков и облачных вычислений значительно снизились, что позволяет использовать эти технологии не только крупному и среднему бизнесу, но и небольшим компаниям и отдельным разработчикам, и пользователям.

Развитие облачных вычислений способствовало развитию параллельной архитектуры программного обеспечения. Приложения разделялись на более мелкие компоненты и «строительные блоки». Распределенные клиент-серверные системы и браузерные веб приложения стали применять модульный принцип, приложения разделяются на функциональные блоки. В то же самое время ряд фреймворков, таких как **Angular** и **React** для **JavaScript**, начали набирать обороты и популярность, что также стандартизировало блоки кода с общими

функциями. В результате появились стандартизированные блоки как на браузерной (**front-end**), так и на серверной составляющей (**back-end**).

Повышение возможностей и качества облачной разработки привело к появления новых форм взаимодействия разработчиков и применения методологии **DevOps**, методологии активного взаимодействия специалистов по разработке со специалистами информационно-технологическому обслуживанию и взаимная интеграция для обеспечения качества продукта (рисунок 2).



Рисунок 2 – **DevOps** методология

Облачная среда способствует развитию подхода **DevOps**. Разработка в облаке для облака означает что аналогичная среда будет использоваться на протяжении всего жизненного цикла продукта что позволяет уменьшить количество возможных ошибок и других проблем в производственной среде.

Использование облачной разработки позволяет создавать более качественное программное обеспечение при меньших затратах на разработку и запуск. Облачная разработка также связана с гибкой методологией (**Agile methodology**) разработки программного обеспечения благодаря использования конвейеров **CI/CD (Continuous Integration/Continuous Delivery)**, т.е. непрерывной интеграции/непрерывной доставки. Выпуск программного обеспечения в виде небольших итераций функциональности заметно снижает затраты и время на разработку за счет избегания разработки ненужных или неэффективных функций. Также облачная разработка позволяет автоматизировать тестирование продукта.

Глобально облачную разработку можно разделить на 2 вида разработки:

1. **Cloud development** (облачная разработка). В общем смысле облачная разработка требует наличия только браузера или онлайн интерфейса соединенных с облачной инфраструктурой.

2. **Cloud-native development** (особый вид облачной разработки). В этом случае идет упор не на **как** создается программное обеспечение, а **что** создается. Сообщества **Kubernetes** и **Cloud Native Computing Foundation (CNCF)** это определяется как разработку, основанную на динамически управляемую разработку программного обеспечения на основе архитектуры микросервисов. Иногда для лучшего понимания термин **cloud-native** заменяется на **container-native development**, буквально означающий «разработка с использованием контейнеров». Стандартная архитектура облачной разработки не использует контейнеры и полагается на саму облачную инфраструктуру для обеспечения масштабируемости и высокой доступности.

Различия между традиционным и облачными методами разработки выглядит следующим образом:

1. Традиционная разработка приложений (**Traditional App Development**):

- виртуальные машины;
- хранилища данных;
- мониторинг безопасности;
- каскадная модель;
- миграции данных.

2. Облачная разработка приложений (**Cloud-Based App Development**):

- «озеро данных» (**data lake**);
- автоматизация безопасности;
- гибкая разработка (**agile**).

3. Особый вид облачной разработки приложений (**Cloud-Native App Development**):

- контейнеры;
- «озеро данных» (**data lake**);
- автоматизация безопасности;
- гибкая разработка (**agile**);
- конвейеры **CI/CD (Continuous Integration/Continuous Delivery)**.

Таким образом, облачная разработка в первую очередь рассматривает управляемую облачную среду как прямую замену локальному серверу. Это уменьшает финансовые вложения в связи с отсутствием необходимости использования локальных серверных мощностей и обеспечивает масштабируемость вычислительной мощности и объема хранилища данных. Но в целом базовая архитектура облачного приложения во многом такая же, как и в традиционными приложениям.

Разработка с использованием контейнеров или особый вид облачной разработки (**cloud-native development**) в полной мере использует облачную среду используя облачные сервисы и функции позволяя унифицировать среды разработки, тестирования и развертывания, что позволяет использовать конвейеры **CI/CD**.

Облачная разработка является будущим стандартом для большинства проектов нового программного обеспечения и миграции стандартного программного обеспечения. Преимущества облачной инфраструктуры и **DevOps** разработка очевидны.

ЛИТЕРАТУРА

1. [www.cloudflare.com URL: https://www.cloudflare.com/learning/cloud/what-is-the-cloud/](https://www.cloudflare.com/learning/cloud/what-is-the-cloud/) (дата обращения: 3.04.2021).
2. <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/#cloud-computing-models> (дата обращения: 28.03.2021).
3. <https://kruschecompany.com URL: 1. https://kruschecompany.com/cloud-development/#:~:text=Cloud%20development%20simply%20means%20writing,to%20a%20cloud%2Dbased%20infrastructure.> (дата обращения: 21.03.2021).

Зинченко Сергей Владимирович – преподаватель кафедры информационных технологий и автоматизированного управления производственными процессами ПГУ им. Т.Г. Шевченко

E-mail: seregagranto@gmail.com

Zinchenko Sergey Vladimirovich - lecturer of the Department of Information Technologies and Automated Control of Production Processes of Shevchenko State University of Pridnestrovie

Балашова Юлия Владимировна – старший преподаватель кафедры интегрированные компьютерные технологии и системы ПГУ им. Т.Г. Шевченко
E-mail:

Balashova Yulia Vladimirovna - senior Lecturer of the Department of Integrated Computer Technologies and Systems of Shevchenko State University of Pridnestrovie