

**МЕТОДИКА ИСПОЛЬЗОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ НА
ОСНОВЕ ИННОВАЦИОННЫХ МЕТОДОВ**

Навшод Абдуллоевич Кудратиллоев

Бекжан Аскарович Ахмедов

преподаватели кафедры «Информатика»

Чирчикский государственный педагогический института

Ташкентской области, Республика Узбекистан

Аннотация: Рассмотрены новая методика в процессах разработки веб-приложения и пути их решения с использованием контейнеризации.

Ключевые слова: архитектура, веб-приложение, разработка.

**METHODS OF USE OF WEB-APPLICATIONS ON THE BASIS OF
INNOVATIVE METHODS**

Navshod Abdulloevich Kudratilloev

Bekjan Askarovich Akhmedov

Teachers of the department «Informatics»

Chirchik State Pedagogical Institute

Republic of Uzbekistan

Annotation: Consideration of new methods in the process of development of web-applications and ways to solve them with the use of containers

Keywords: architecture, web application, development, cluster, education.

Контейнеризация - это процесс распространения и развёртывания приложений портируемым и предсказуемым образом. Это достигается путём упаковывания компонентов и их зависимостей в стандартные, изолированные и лёгкие окружения (process environments), называемые контейнерами. В настоящее время многие организации заинтересованы в создании приложений и сервисов, которые могут быть легко развёрнуты на распределенных системах, позволяя системе легко масштабироваться и оставаться работоспособной при отказе отдельных машин или приложения.

Docker - платформа для контейнеризации, разработанная для упрощения и стандартизации развертывания в разных окружениях, которая в значительной степени помогла распространению данной практики проектирования и управления сервисами. Разработано большое количество программного обеспечения, работающего на основе данной экосистемы распределённого управления контейнерами.

Без стандартизации окружений, в котором запускается проект во время разработки и эксплуатации, каждое из них (локальное у разработчика, тестовое и рабочее окружения) приходится настраивать по отдельности. Причем не всегда этим занимается один и тот же человек. Чтобы подготовить отдельное окружение для запуска проекта, необходимо произвести его настройку:

1. Установить компилятор или интерпретатор языка программирования.
2. Установить базу данных.
3. Установить другие системные утилиты.
4. Клонировать репозиторий.
5. Заставить все компоненты работать вместе.

Причем не всегда получается сделать минимальной разницу между всеми окружениями. Из-за этого может возникнуть такая ситуация, когда на рабочем окружении код не работает, а разработчик, написавший его, утверждает, что у него «локально все работает».

На пути продвижения вашего приложения через цикл разработки и, в конечном итоге, в эксплуатацию зачастую встречается множество препятствий. Помимо работы по подготовке приложения к работе в различных условиях, Вы также можете столкнуться с проблемами отслеживания зависимостей, масштабирования приложения и обновлений отдельных компонентов, не затрагивающих непосредственно само приложение.

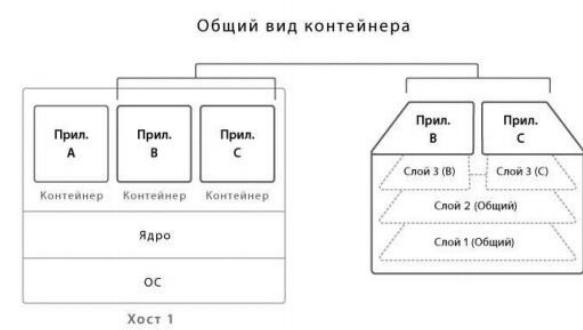
Docker-контейнеризация и сервис-ориентированное проектирование пытаются решить многие из этих проблем. Приложения могут быть разбиты

на управляемые функциональные компоненты, индивидуально упакованные вместе со всеми своими зависимостями, а затем легко развернуты на нестандартной архитектуре. Это также упрощает масштабирование и обновление компонентов.

На сегодняшний день Docker является самым распространённым программным обеспечением для контейнеризации. Хотя существуют и другие системы контейнеризации, Docker делает создание и управление контейнерами простым и легко интегрируется с многими open-source проектами.

Рис. 1. Устройство контейнера в общем виде

На картинке выше Вы можете увидеть (в упрощенном виде), как контейнеры соотносятся с хост-системой. Контейнеры изолируют отдельные приложения и используют ресурсы операционной системы с помощью Docker. В развернутом виде на картинке справа можно видеть, что контейнеры построены при помощи "слоев", при этом несколько контейнеров могут иметь общие нижние слои, уменьшая использование ресурсов.



Обычно на этапе проектирования приложения или сервиса, в котором будет использоваться Docker, лучше всего разнести функциональность по разным контейнерам, такой подход называется сервис-ориентированной архитектурой. Он дает возможность в последствии легко масштабировать или обновлять компоненты независимо друг от друга. Наличие такой гибкости - одна из множества причин, по которым люди заинтересованы в Docker для разработки и развертывания приложений.

Контейнеры несут в себе много привлекательных преимуществ как для разработчиков, так и для системных администраторов. Некоторые из наиболее заманчивых преимуществ перечислены ниже.

- Простота работы с версиями: Docker-файлы могут быть сохранены в системе контроля версий для отслеживания изменений и "откатывания" ошибок.
- Предсказуемость: Сборка образов из Docker-файла помогает избавиться от человеческих ошибок в процессе создания образов.
- Контролируемость: Если Вы планируете делиться своим образом, обычно хорошим тоном является предоставление Docker-файла, который создает данный образ, чтобы другие пользователи могли проконтролировать процесс. Таким образом Вы, по сути, предоставляете историю предпринятых для построения образа шагов.
- Гибкость: Создание образов из Docker-файла позволяет переопределить настройки, которые заданы по умолчанию в интерактивном режиме. Это значит, что Вам надо производить меньше действий, чтобы получить правильно функционирующий образ.

REFERENCES:

1. Гулбаев, Н. А., Кудратиллоев, Н. А. (2020). Состояние проблем управления систем с рассредоточенными объектами (на примере электрических сетей). *Science and World*, 6(82), 29-32.
2. Гулбаев, Н. А., Кудратиллоев, Н. А. (2020). Моделирование и управление территориально-распределенными системами. *Science and World*, 6(82), 25-28.
3. Гулбаев, Н. А., Кудратиллоев, Н. А. (2020). Модели упорядочивания структур управления систем с рассредоточенными объектами. Евразийский Союз Ученых (ЕСУ), 6(75), 46-48.

4. Ахмедов, Б. А. (2021). Задачи обеспечения надежности кластерных систем в непрерывной образовательной среде. *Eurasian Education Science and Innovation Journal*, 1(22), 15-19.
5. Akhmedov, B. A., Makhkamova, M. U., Aydarov, E. B., Rizayev, O. B. (2020). Trends in the use of the pedagogical cluster to improve the quality of information technology lessons. Экономика и социум, 12(79), 802-804.
6. Akhmedov, B. A., Majidov, J. M., Narimbetova, Z. A., Kuralov, Yu. A. (2020). Active interactive and distance forms of the cluster method of learning in development of higher education. Экономика и социум, 12(79), 805-808.
7. Akhmedov, B. A., Eshnazarova, M. Yu., Rustamov, U. R., Xudoyberdiyev, R. F. (2020). Cluster method of using mobile applications in the education process. Экономика и социум, 12(79), 809-811.
8. Akhmedov, B. A., Kuchkarov, Sh. F., (2020). Cluster methods of learning english using information technology. Scientific Progress, 1(2), 40-43.
9. Akhmedov, B. A. (2021). Development of network shell for organization of processes of safe communication of data in pedagogical institutions. Scientific progress, 1(3), 113-117.
10. Ахмедов, Б. А., Шайхисламов, Н., Мадалимов, Т., Махмудов, К. (2021). Smart технологияси ва ундан таълимда тизимида кластерли фойдаланиш имкониятлари. Scientific progress, 1(3), 102-112.
11. Akhmedov, B. A., Majidov, J. M. (2021). Practical ways to learn and use the educational cluster. Экономика и социум, 2(81).
12. Akhmedov, B. A. (2021). Cluster methods for the development of thinking of students of informatics. Academy, 3(66), 13-14.
13. Камиловна, А. И. (2020). Лойиҳалаш фаолиятининг босқичлари лойиҳага ёндашув моҳияти ва мақсадлар уйғунлиги. Science and Education, 1(2), 18-26. 22.

14. Камиловна, А. И. (2020). Глобаллашув шароитида ёшларни ватанпарварлик руҳида тарбиялашнинг долзарбилиги. *Science and Education*, 1(2), 204-211.