

13. Bourdeau C., Billauxand D., Chraibi A.F. Thermo-hydro-mechanical analysis of a DAM using FLAC3D software. FKAC/DEM, Minneapolis, 2008.

УДК 004.4

Чернов Андрей Владимирович

Chernov Andrey Vladimirovich

Д.т.н., профессор

Dr.Tech.Sc., Professor

Коновалов Никита Сергеевич

Konovarov Nikita Sergeevich

Ростовский государственный университет путей сообщения

Rostov State Transport University

Побойкина Алина Олеговна

Poboykina Alina Olegovna

Студент

Student

Донской государственный технический университет

Don State Technical University

СТРАТЕГИЯ РАЗВЕРТЫВАНИЯ МИКРОСЕРВИСОВ В ОБЛАКЕ

STRATEGY FOR DEPLOYING MICROSERVICES IN THE CLOUD

Аннотация: В данной статье рассматриваются способы развертывания микросервисов в облаке. Целью данной статьи является выбор наиболее подходящего способа развертывания микросервиса в облаке. В результате исследования рассмотрены различные стратегии развертывания микросервисов и приведены детальные примеры возможного развертывания с выделением положительных и отрицательных сторон.

Abstract: This article discusses how to deploy microservices in the cloud. The purpose of this article is to choose the most appropriate way to deploy a microservice in the cloud. As a result of the study, various strategies for deploying microservices are considered and detailed examples of possible deployments are given, highlighting the positive and negative aspects.

Ключевые слова: микросервисная архитектура, развертывание, облачное развертывание, стратегия развертывания.

Keywords: microservice architecture, deployment, cloud deployment, deployment strategy.

Введение. Приложение, которое содержит в себе микросервисы, состоит из множества сервисов, написанных на разных языках и фреймворках. Каждый из микросервисов представляет собой отдельное приложение со своим особенным окружением и условиями к развертыванию, мониторингу, ресурсам [1]. К примеру, появляется надобность в нескольких копиях сервиса в зависимости от его нагруженности и востребованности, также каждому сервису необходимо предоставить соответствующее количество ресурсов. Сложнее всего то, что сервисы должны быстро устанавливаться, а также быть надежными.

Существует несколько шаблонов развертывания, которые представлены на рис. 1 [2].

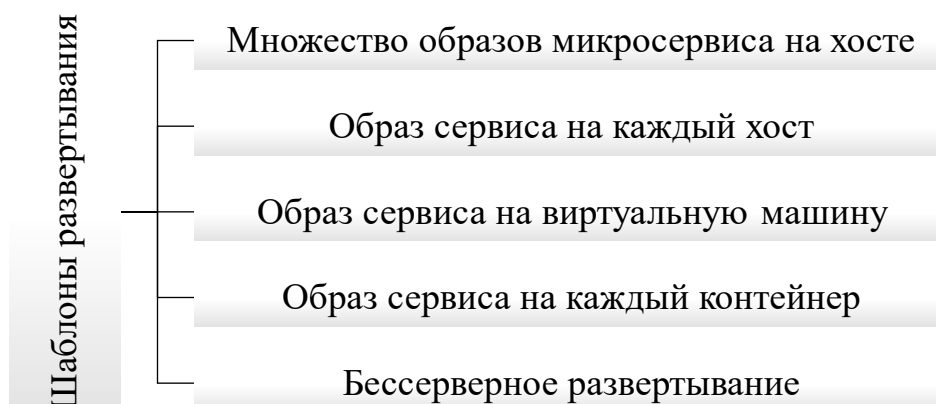


Рис. 1. Виды шаблонов развертывания микросервисов

Множество образов микросервиса на хосте. В данном подходе используется шаблон развертывания нескольких экземпляров на хосте. При использовании данного подхода, на предоставляемых облачных хостах, запускается несколько экземпляров сервисов на каждом [3, 5]. Плюсом является то, что при хорошей балансировке сервисы будут доступны для большего количества клиентов и в случае отказа одного из сервисов, работа не

остановится. Минусом является то, что не каждый сервис возможно таким образом разбить (к примеру, сервис работает с глобальной базой и обновляет какой-либо компонент по определенным условиям, если два сервиса одновременно будут делать противоположную операцию, это может привести к неправильному результату).

Образ сервиса на каждый хост. При использовании данного шаблона каждый образ сервиса запускается на отдельном хосте, что позволяет изолировать экземпляры [4]. Плюсом данного подхода является то, что в случае если один из сервисов расходует все ресурсы системы, то остальные образы на других хостах от этого не страдают и могут продолжать свою работу. Существенным минусом является затратное содержание нескольких хостов и контроль их состояния, необходимость в отдельном специалисте по серверу для конфигурации.

Образ сервиса на виртуальную машину. Используя данный шаблон, каждый сервис упаковывается в отдельную виртуальную машину (например, как yandex cloud). Каждый образ сервиса – это виртуальная машина, которая запускается из конкретной сборки виртуальной машины [3]. Данный подход в основном используют крупные компании для хранения изолированных сборок. Положительным моментом данного подхода является то, что нет необходимости тратить время на настройку окружения, так как это доступно из панели конфигураций, а также в данном шаблоне, как и в шаблоне образа микросервиса на каждый хост, ресурсы выделяются под конкретную виртуальную машину и не могут влиять на другие виртуальные машины. Отрицательной же стороной является то, что ресурсы используются не совсем правильно, расходы идут на всю виртуальную машину, а именно операционную систему. Также как правило имеются фиксированные шаблоны системы, которые могут быть избыточны.

Образ сервиса на каждый контейнер. Данный шаблон развертывания микросервиса является простым, но при этом имеет положительные стороны развертывания такие же, как на виртуальных машинах. При использовании данного шаблона, во время каждой сборки образ сервиса запускается в

собственном контейнере. Контейнеры – это механизм, реализующий виртуализацию на уровне операционной системы, состоит из процессов, исполняющихся в своем виртуальном окружении. С точки зрения процессов и окружения, это отдельная виртуальная система, которая имеет свою корневую файловую структуру, порты и пространство имен. В данном шаблоне имеется возможность ограничивать ресурсы контейнера, а именно оперативную память, дисковое пространство, а также загруженность `cpu`. Наиболее известной контейнерной технологией является `docker` [4]. При применении данного шаблон создается образ `docker` контейнера с необходимым окружением для нужного приложения. После создания конфигурационного файла образа, появляется возможность запустить несколько образов на физических или виртуальных хостах. Плюсами данного шаблона является то, что имеется возможность простого наблюдения за ресурсами, которые расходует каждый образ, вдобавок возможность изменять допустимые объемы выделяемых ресурсов. Из минусов можно обозначить, что контейнеры используют совместно ядро системы хоста, а также сложность администрирования, если не применяются специализированные решения.

Бессерверное развертывание. Основной особенностью данного шаблона является то, что основу составляют функции, исполняющие конкретную задачу с возможностью подписаться на конкретные события в облаке. Этот подход позволяет запускать приложения в безопасном, отказоустойчивом и масштабируемом окружении, без создания и обслуживания виртуальных машин [5, 6]. Из отрицательных сторон неизвестность места запуска функции, отсутствие структурированности приложения, так как каждая функция, это отдельный объект.

Заключение. Развертывание микросервисов в облаке не такая тривиальная задача, как может показаться. При выборе определённого подхода следует четко оценить все положительные и отрицательные стороны, а также сложность администрирования, ведь каждый микросервис, это отдельное приложение со своими особенностями, на которые необходимо обращать внимание.

Библиографический список:

1. Штефуряк, А. С. Грамотный выбор стратегии развёртывания микросервисного программного продукта / А. С. Штефуряк, Т. С. Яницкая // Молодой ученый. 2020. № 43 (333). С. 19-21.
2. Маркелов А. А. Введение в технологии контейнеров и Kubernetes – М.: ДМК Пресс, 2019.
3. Михайлов К.И. Анализ программных средств мониторинга высоконагруженных облачных информационных систем // Системный анализ в проектировании и управлении. 2019. С. 359-362.
4. Грузин Н.А. Сравнение Docker контейнеров и виртуальных машин, а также облачных платформ AWS и Azure // MODERN SCIENCE. 2020. №2-1. С. 351-355.
5. Надейкина Л.А., Черкасова Н.И. Распределенные системы, построенные на базе микросервисной архитектуры // Инновационные, информационные и коммуникационные технологии. 2019. №1. С. 300-304.
6. Артамонов Ю.С., Востокин С.В. Разработка распределенных приложений сбора и анализа данных на базе микросервисной архитектуры // Известия Самарского научного центра Российской академии наук. 2016. №4(4). С. 688-693.