

Программная инженерия, лекция 6: Микросервисная архитектура

Пожидаев М. С.

10 октября 2022 г.

Микросервисная архитектура

Контроллер, брокер и не только

Блоки рассматриваемой архитектуры:

1. *Контроллер*: центральный узел, получающий команды от внешних клиентских соединений.
2. *Брокер сообщений*: диспетчер обмена сообщениями между контроллером и остальными контейнерами.
3. *Микросервисы*: динамически изменяемое множество одноранговых контейнеров разных типов для выполнения работ, необходимых для обслуживания запросов.

Микросервис

Основные черты

1. Слабая связь с остальными микросервисами.
2. Изолирование фактической технологии внутри микросервиса (каждый микросервис разрабатывается своей командой).
3. Коммуникация путём сетевого взаимодействия (предпочтительно через HTTP).
4. Оформление в виде контейнера.
5. Могут находиться территориально распределённо.

Проектирование

Как проектировать микросервисную архитектуру

1. Формализация ясного функционального назначения микросервисов.
2. Максимальное устранение дублирования функциональности между микросервисами.
3. Фиксация интерфейсов и связей между микросервисами.
4. Фиксация конфигурации для развёртывания и требования к ней.

Контейнерная виртуализация

Основные черты контейнерной виртуализации

1. Код выполняется нативно.
2. В гостевых системах ядро общее с хост-системой.
3. Каждый контейнер имеет свой IP-адрес.
4. Лимитируется память, процессорное время и пр.

Недостатки

Минусы контейнерной виртуализации

При сплошных плюсах технологии контейнерной виртуализации есть и некоторые минусы:

1. Слишком быстро копится overhead использования дискового пространства.
2. Сильная привязка к ядру хост-системы.
3. На Windows (равно как и в других ОС вообще) выполняется в виртуальной машине, что сводит на нет полезный эффект.
4. Не эмулирует недоступное оборудование.

Технологии

Реализация контейнерной виртуализации

Пространства имён (namespace)

Одна копия Linux может разделять на пространства имён следующие подсистемы: файловые системы, идентификаторы процессов, сетевые атрибуты, элементы межпроцессного взаимодействия, идентификаторы пользователей.

Контрольные группы(cgroups)

Позволяют лимитировать интенсивность использования таких ресурсов как: память, сетевой обмен, ввод/вывод, процессорное время.

Docker

От компании Google

Docker — система управления контейнерами на базе пространств имён и контрольных групп Linux.

1. Автоматически конфигурирует сеть для гостевых систем и хост-системы.
2. Использует каскадное объединение томов хранения файлов.
3. Поддерживает публичный репозиторий образов контейнеров.
4. Написана на языке Go.

Прямо на ходу

Задача «горячего» обновления

Требования к решению задачи «горячих» обновлений (rolling updates):

1. Контейнеры, не являющиеся контроллером и брокером, периодически получают обновления. Чем чаще, тем лучше.
2. Работу сервиса останавливать нельзя, пользователи должны наблюдать непрерывное функционирование с нормальной обработкой запросов.
3. Близкой является задача динамического изменения набора контейнеров-работников для увеличения или уменьшения задействованных вычислительных мощностей.

Порядок обновления

Действия при «горячем» обновлении

1. Механизм реализуется в непосредственной связи с используемым брокером, выполняющим обслуживание межконтейнерного взаимодействия. Именно он чаще всего хранит информацию о конфигурации системы.
2. Инструмент оркестрации получает команду изменить конфигурацию системы, определяет набор контейнеров, которые должны быть остановлены.
3. Контейнеры получают сигнал завершить свою работу (SIGTERM), после чего уведомляют брокер о своём новом статусе.
4. Контейнеры завершают выполнение тех задач, которые были в работе на момент получения сигнала завершить работу.
5. Контейнеры либо добровольно завершают свою работу, либо их убивают по timeout при помощи SIGKILL.

Оркестрация контейнеров

Когда контейнеров становится много, то управление ими требует применение дополнительных инструментов.

Существует ряд утилит, которые решают задачу так называемой оркестрации контейнеров: запуск необходимого количества контейнеров и поддержания их работоспособности. Среди них наиболее популярны:

- ▶ Kubernetes;
- ▶ Docker Swarm.

Спасибо за внимание!

Всё о курсе: <https://marigostra.ru/materials/engineering.html>

E-mail: msp@luwrain.org

Канал в Телеграм: <https://t.me/MarigostraRu>