

“Управление операционными системами с использованием микросервисной архитектуры в РЕД АДМ”

Грачев Дмитрий Львович, ООО «РЕД СОФТ»

dmitry.grachev@red-soft.ru

Введение

«РЕД АДМ» — инструмент централизованного управления и администрирования информационной инфраструктуры.

В процессе своего развития спектр задач которые должен решать РЕД АДМ постоянно растет. К примеру, РЕД АДМ позволяет решать задачи развертывания и администрирования контроллера домена, управления учетными записями в домене, настройки общих сетевых ресурсов, централизованной установки операционных систем, журналирования и др.

С учетом гетерогенности задач, которые помогает решать РЕД АДМ, отделу разработки РЕД АДМ также приходится сталкиваться с разноплановыми задачами, поэтому модернизация архитектуры приложения для поддержки уже написанного кода и интеграции нового, стала актуальной и необходимой задачей, не только чтобы изолировать компоненты РЕД АДМ, увеличив тем самым консистентность подсистем, из которых он состоит, но также уменьшить связанность кода, а значит количество времени необходимое для внедрения нового функционала и уменьшение вероятности того, что он повлияет на существующий.

Задачи

- определить целесообразность перехода с монолитной на микросервисную архитектуру
- установить обеспечивает ли микросервисная архитектура поддержку разных платформ
- определить стоит ли менять архитектуру всего приложения или же только новые подсистемы будут писаться с использованием нового архитектурного подхода
- провести необходимую подготовительную работу существующего монолита, не затронув уже написанный функционал
- провести мягкую (постепенную) миграцию от монолита к микросервисной архитектуре

Выводы

Миграция от монолитной к микросервисной архитектуре - это, несомненно усложнение архитектуры приложения. Оно ведет к усложнению:

- тестирования
- отладки
- развертывания
- обновления

Однако, по мере увеличения команды разработки, кодовой базы, количества и разноплановости задач, которые необходимо решать, - цена, в виде усложнения архитектуры, позволяет, на определенном этапе:

- упростить администрирование и поддержку разных платформ

- сделать проще процесс разработки
- уменьшить количество возможных ошибок, благодаря уменьшению связанности кода
- быстрее начать писать код и внедрять новый функционал, в том числе, - новому сотруднику
- увеличить изолированность компонентов программы (подсистем), тем самым повысить ее надежность