

ПЛАНЫ ПО РАЗВИТИЮ ИНТЕГРАЦИОННЫХ ПРОДУКТОВ

JBoss Middleware

Дмитрий Володин
Системный архитектор,
Apache Camel Community Contributor

Москва, 22 Сентября 2017

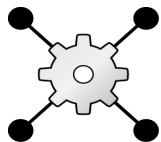
ЭВОЛЮЦИЯ ТЕХНОЛОГИЙ ИНТЕГРАЦИИ

	1970	1980	1990	2000	2010
Технологии	Общие ресурсы на мэйнфреймах ISAM VSAM IDMS	Точка-точка Проприетарные TCP сокет, FTP	Концентратор Брокеры сообщений, CORBA	Оркестрация SOA Web-сервисы XML <i>Открытый исходный код</i>	Медиация сервисов REST JSON API's
Проблемы бизнеса	Согласованность данных	Согласованность данных Консолидированная отчетность	Предыдущие, плюс Электронные заказы Управление кейсами Сквозные процессы	Предыдущие, плюс Многошаговые и управляемые внешними системами процессы Композитные и мобильные приложения	Предыдущие, плюс Миграция в "облака" API как бизнес сервис "Интеграция" с AI

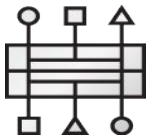
ИЗМЕНЕНИЕ ЛАНДШАФТА ДЛЯ ИНТЕГРАЦИИ

ИТ предприятий претерпевают серьезные изменения. Чтобы оставаться конкурентоспособными, им нужна платформа интеграции, способная поддерживать архитектуру текущего и **следующего** поколений.

Конечные точки



Web-сервисы

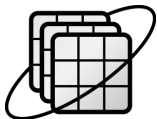


API

Архитектура

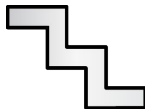


Монолитная

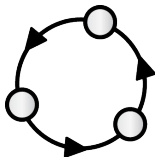


Микросервисная

Процесс разработки



Модель “Водопад”

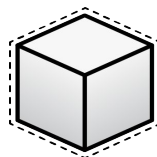


CI/CD

Развертывание

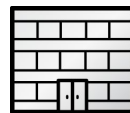


Сервера/VM



Контейнеры

Инфраструктура



ЦОД



Облако

ТРИ СОСТАВЛЯЮЩИЕ ГИБКОЙ ИНТЕГРАЦИИ

Базовые возможности, необходимые современным предприятиям

Распределенная интеграция

Легковесные, основанные на шаблонах, ориентированные на события, механизмы подключения

Гибкость

Контейнеры

Контейнерное масштабирование, инкапсуляция и управление системами

Масштабируемость

API

Точно определенные и документируемые, повторно используемые, управляемые конечные точки

Повторное использование

ТЕХНОЛОГИИ ДЛЯ НАДЕЖНОЙ ОСНОВЫ

Распределенная интеграция

Легковесные, основанные на шаблонах, ориентированные на события, механизмы подключения

**RED HAT® JBOSS®
FUSE**

**RED HAT® JBOSS®
A-MQ**

Контейнеры

Контейнерное масштабирование, инкапсуляция и управление системами

 **RED HAT®
OPENSIFT**

API

Точно определенные и документируемые, повторно используемые, управляемые конечные точки

 **3scale**
BY RED HAT®

JBoss AMQ 7

JBOSS AMQ 7

**Стандартные
протоколы**

**Единые
инструменты
управления**

Гибкая и основанная на стандартах система обмена сообщениями для предприятия, облака или “интернета вещей” (IoT)

Брокер

- Новое ядро брокера с современной асинхронной архитектурой
- Улучшенная производительность и масштабируемость

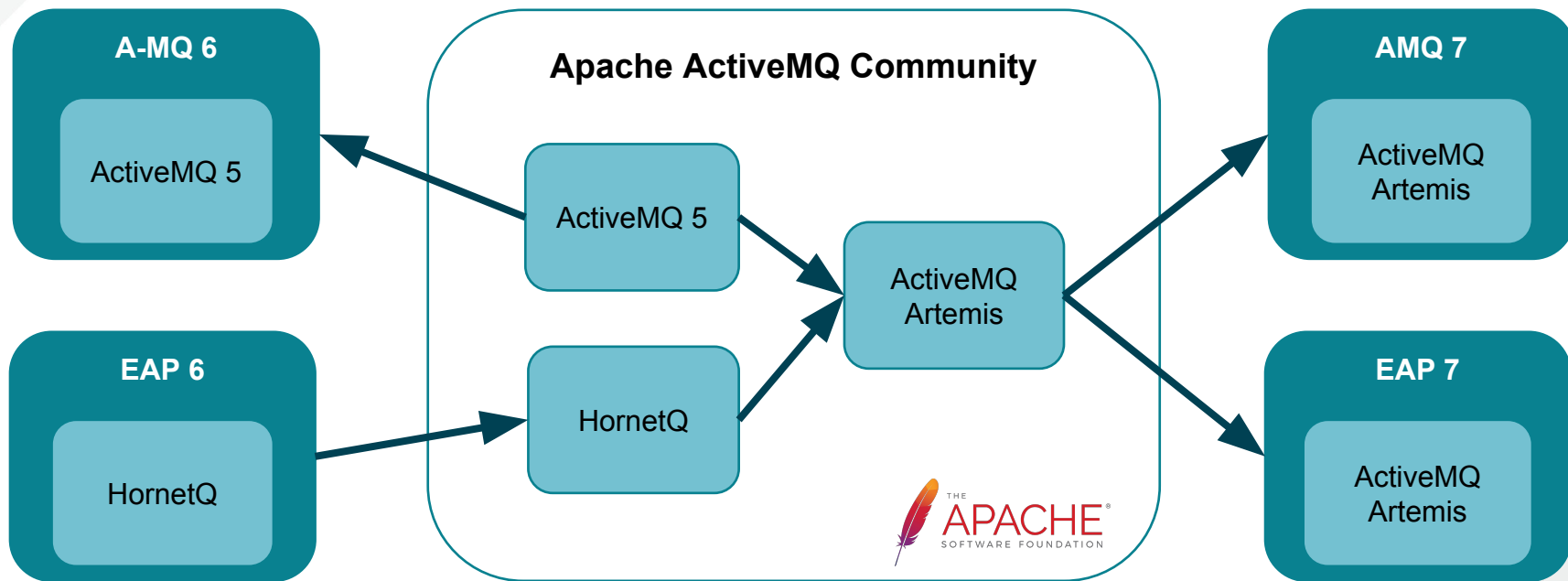
Interconnect

- Роутер сообщений
- Высокопроизводительный прямой обмен сообщениями
- Распределенная система обмена сообщениями

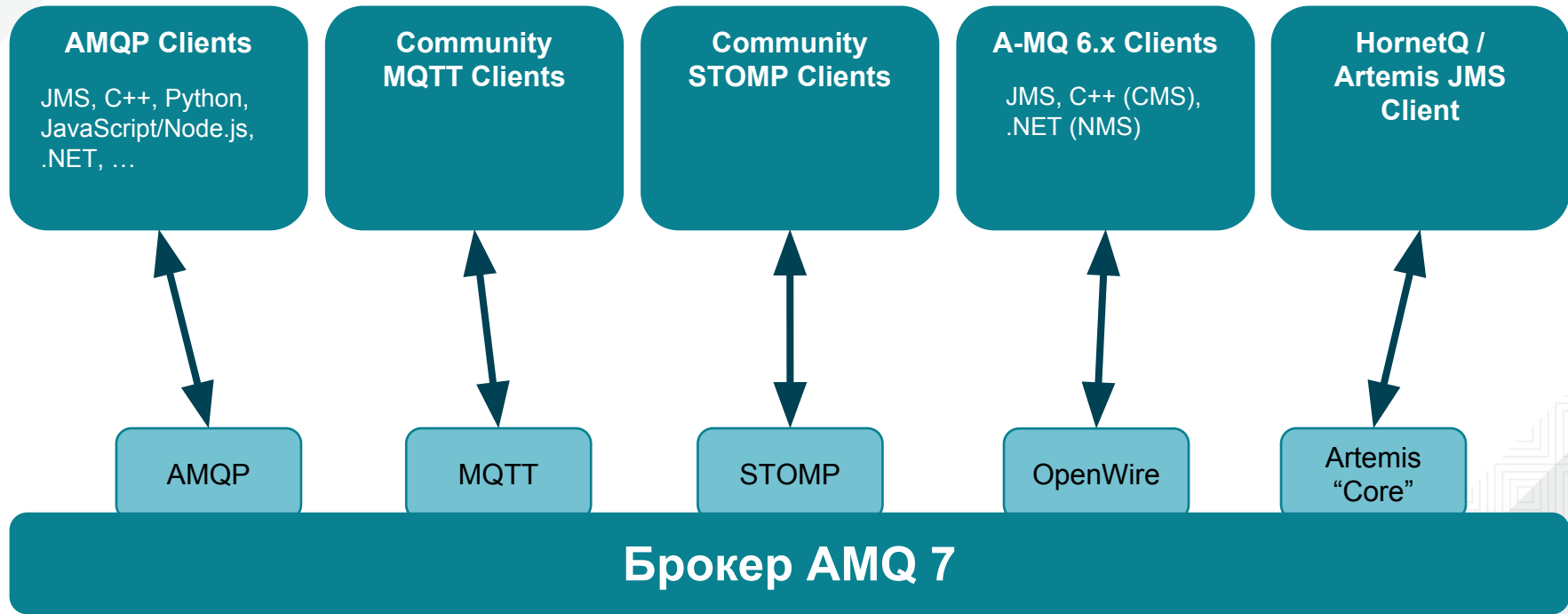
Клиенты

- Новый набор клиентов AMQP 1.0
- Поддержка форматов AMQ 6.x и HornetQ клиентов для обратной совместимости

КОНСОЛИДАЦИЯ НЕСКОЛЬКИХ РЕШЕНИЙ



ПОДДЕРЖИВАЕМЫЕ ПРОТОКОЛЫ И КЛИЕНТЫ



ОБРАБОТКА СООБЩЕНИЙ КАК СЕРВИС (MaaS)

AMQ 7 для OpenShift

- Позволяет приложениям, работающим в OpenShift, использовать подсистему обработки сообщений
- Выставляет сервисы, развернутые в OpenShift, доступными для внешних приложений
- Обеспечивает взаимодействие на уровне сообщений между различными приложениями/проектами, работающими в OpenShift
- Передача сообщений между публичными облачными и локальными сервисами

JBoss FUSE

К ЧЕМУ МЫ ДОЛЖНЫ БЫТЬ ГОТОВЫ?



FUSE 6.3

ОБНОВЛЕНИЕ

Camel 2.17
23 новых компонентов Camel
CXF 3.1
Jetty 9

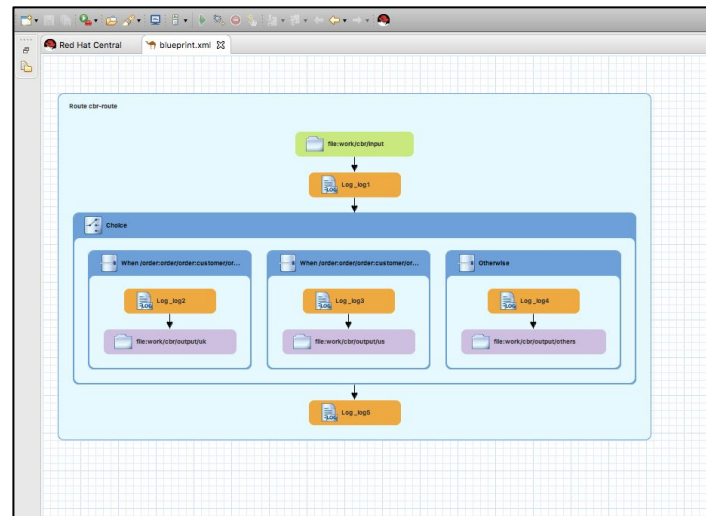
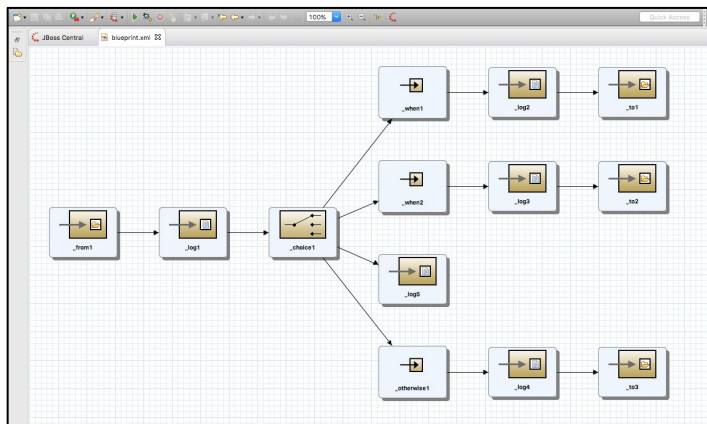
УДОБСТВО

Обновленный редактор потоков
Новые примеры использования
Унификация Maven репозитория

СТАБИЛЬНОСТЬ

Исправлено 2500 проблем во всех
подпроектах
Исправлено 1191 проблем в продукте
204 проблемы исправлено в Fabric8 v1

ОБНОВЛЕННЫЙ РЕДАКТОР ПОТОКОВ



ЧТО ТАКОЕ FUSE INTEGRATION SERVICES (FIS)?

Fuse Integration Services (FIS) - это вариант распространения JBoss Fuse, который обеспечивает поддержку инструментов и механизмов для создания контейнеризированных интеграционных сервисов на платформе OpenShift, включая:

- Образ контейнера в формате Docker
- Инструменты для создания, разработки и сборки контейнеризированных приложений Fuse
- Шаблоны для большинства интеграционных сценариев
- Непосредственная интеграция с Kubernetes для поиска сервисов, их кластеризации и конфигурации
- Все построено на базовых технологиях JBoss Fuse

ЛЕГКОВЕСНАЯ СРЕДА ИСПОЛНЕНИЯ ИНТЕГРАЦИЙ

Образы контейнеров в формате Docker для Karaf и Spring Boot обеспечивают основу для построения контейнеризированных интеграционных сервисов.

SPRING BOOT RUNTIME

Разработка кода и конфигурация

Специфические для приложения
зависимости

Spring Boot

fis-java-openshift

- Простая конфигурация, bean-контейнер
- Поддержка Spring и Java DSL для Apache Camel
- Автоматическое связывание (Autowired)
- Основан на Camel 2.18

KARAF RUNTIME

Разработка кода и конфигурация

Специфические для приложения
зависимости

Apache Karaf

fis-karaf-openshift

- OSGi контейнер на базе Karaf
- Поддерживает Blueprint DSL для Apache Camel
- Выровнен по версиям с Fuse 6.3 для более простого перехода от развертывания Fuse на базе Fabric8 v1 к варианту развертывания на OpenShift

FUSE 7.x

Три основных аспекта развития продукта



Fuse Standalone



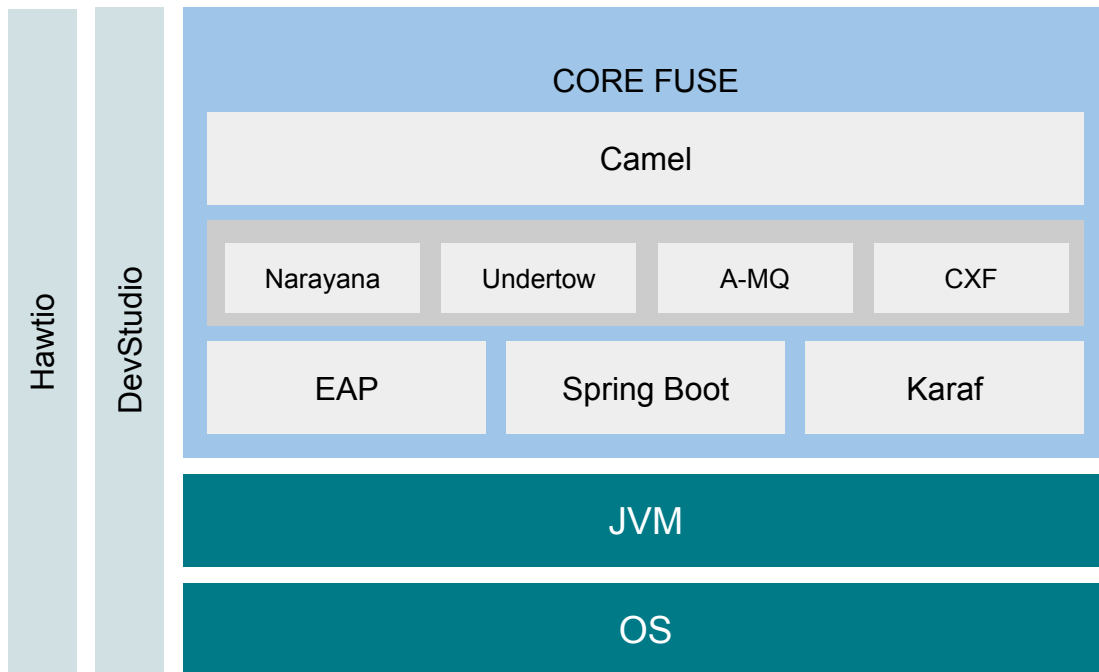
Fuse Integration Services



Fuse iPaaS



FUSE STANDALONE



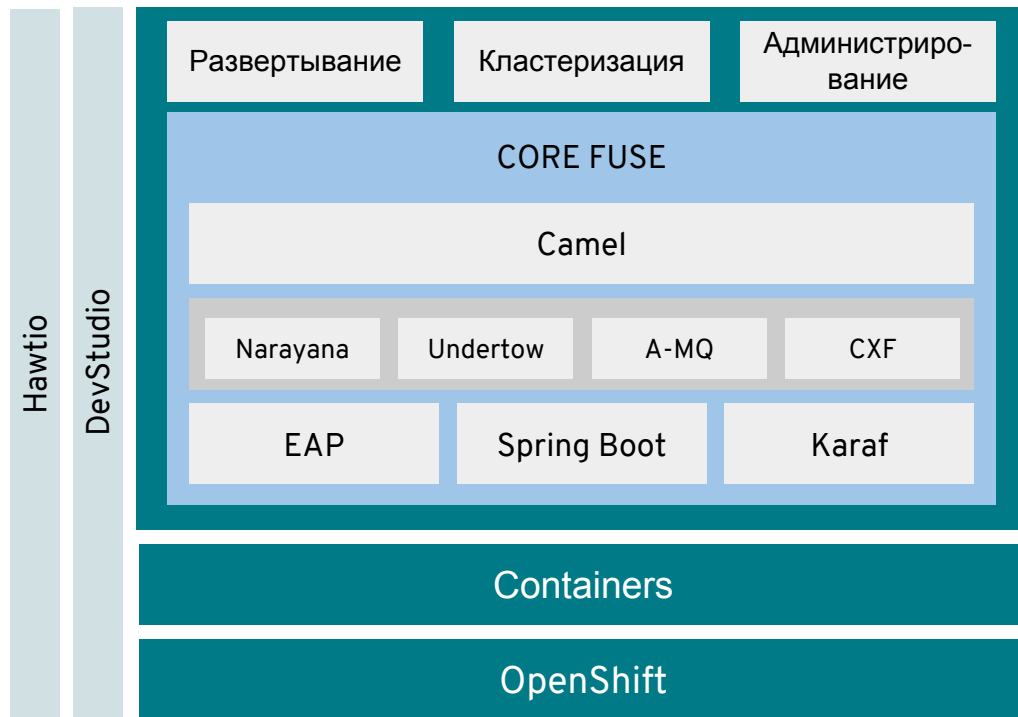


FUSE STANDALONE

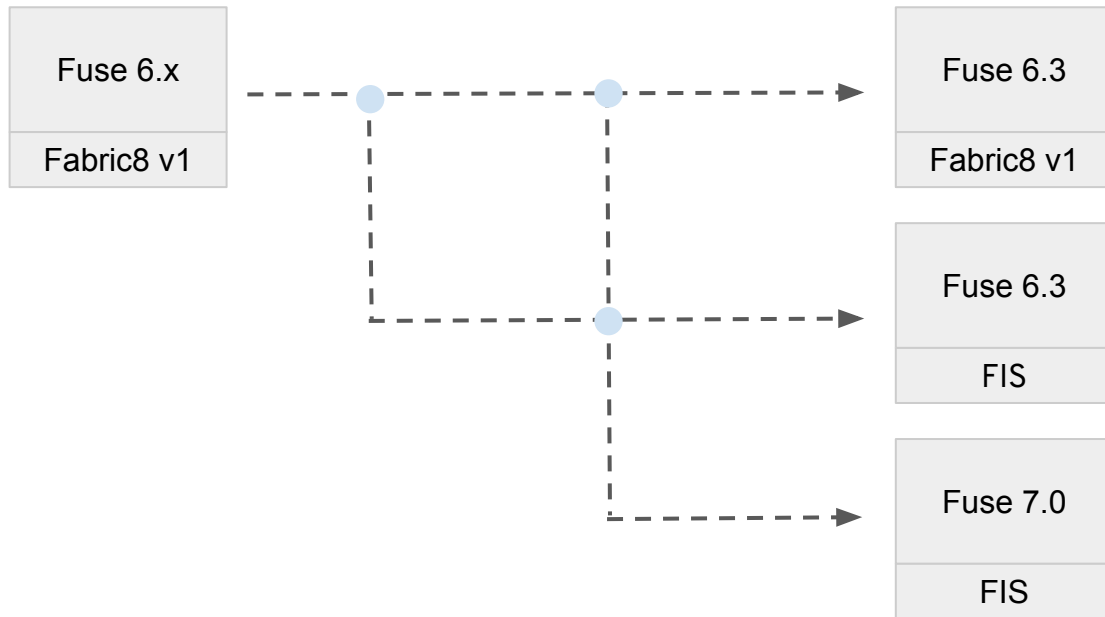
- Camel 2.20
- Karaf 4
- Поддержка Spring Boot
- Поддержка EAP 7.x
- Менеджер транзакций Narayana
 - Geronimo не поддерживается
- Web-контейнер Undertow
 - Jetty не поддерживается
- Сертификация с AMQ 6/7
- Дополнительные компоненты для EAP 7
- SwitchYard больше не поддерживается
- Интеграция с фокусом на API
 - Редактор REST DSL
 - SOAP <-> REST
 - Интеграция с Red Hat 3scale



FUSE INTEGRATION SERVICES



ПЕРЕХОД Fuse 6.x -> Fuse 7.0



Оставайтесь на той же версии
Fuse (End of Maintenance
Support - Jan 2022)

Используйте FIS 2.0, когда
будете готовы

Ждите Fuse 7.0

FUSE iPaaS

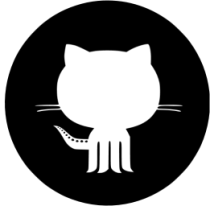


...CKOPO...



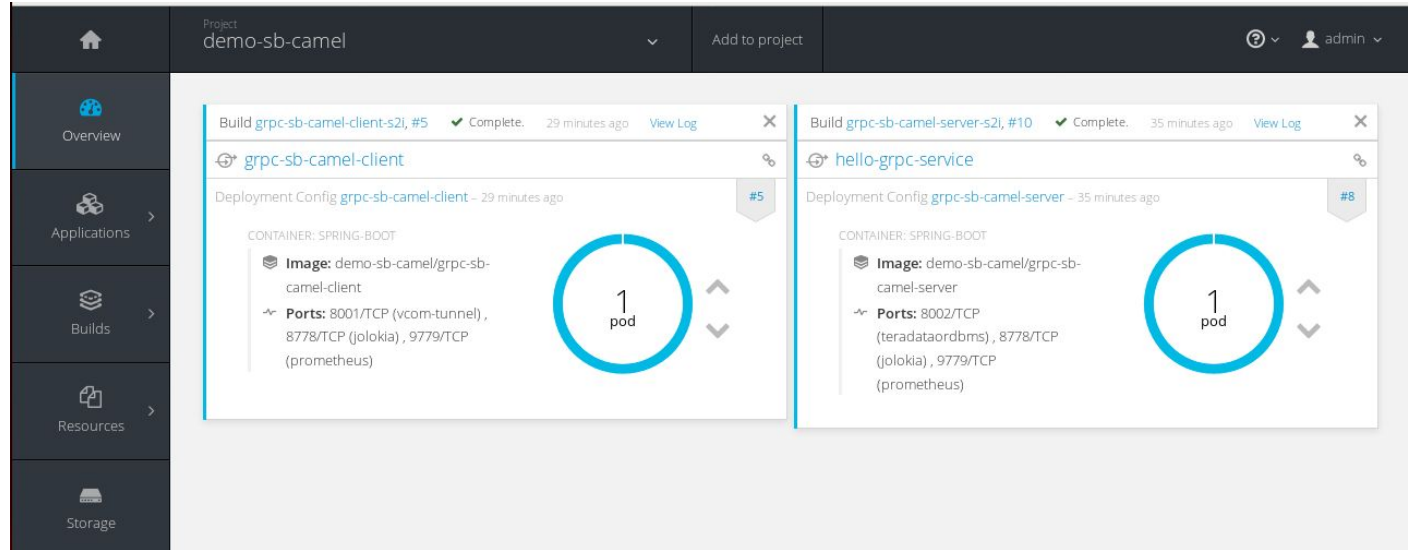
redhat.®

ДЕМОНСТРАЦИЯ



Red Hat Forum Moscow 2017 Demo Camel client and server

<https://github.com/dmvolod/rh-forum-moscow-2017>

The screenshot shows the OpenShift web console interface. At the top, a dark header bar contains a home icon, the project name 'demo-sb-camel', an 'Add to project' button, and user information 'admin'. On the left, a sidebar menu includes 'Overview' (selected), 'Applications', 'Builds', 'Resources', and 'Storage'. The main content area displays two deployment configuration cards side-by-side. The left card is for 'grpc-sb-camel-client', showing it is 'Complete' 29 minutes ago. It details the container 'demo-sb-camel/grpc-sb-camel-client' with ports 8001/TCP, 8778/TCP, and 9779/TCP, and indicates '1 pod' is running. The right card is for 'hello-grpc-service', also 'Complete' 35 minutes ago, with the container 'demo-sb-camel/grpc-sb-camel-server' and the same port configuration, also showing '1 pod' running. Both cards include a 'View Log' link.



RED HAT **FORUM**

Europe, Middle East & Africa