

### ОПТИМИЗАЦИЯ РАБОТЫ JVM В ОБЛАКЕ

#### «ПЛАТФОРМА КАК CEPBИС» JELASTIC

Масштабируемая, безопасная и гибкая облачная платформа для разработки новых и развертывания существующих бизнес-приложений корпоративных и

• Масштабируемая инфраструктура разработи и ветирования ПО

• Операционные системы различных производителей

• СУБД, middleware, языковые платформы и фрэймворки

• Быстрое создание и вывод в эксп<del>луатаци</del> сервисов

• Экспертиза в автоматизации сложных DevOpsпроцессов



#### ПЛАТФОРМА ДЛЯ РАЗРАБОТКИ ИННОВАЦИОННОГО ПО

- Разворачивание существующих и разработка новых бизнес-приложений
- Упрощение и стандартизация процессов разработки ПО
- Значительное ускорение и удешевление разработки и обслуживания
  - комплексного ПО
- Автоматическое масштабиро кластеризация



BUILD





- Большое количество поддержива технологий
- Сервер приложений и middleware как сервис (PaaS) за данных как сервис (DBaaS)

#### ТЕХНОЛОГИЯ КОМПАНИИ

Технология Jelastic испытана множеством высоконагруженных приложений по всему миру. Наше программное обеспечение используется 40+ провайдерами облачных услуг, корпоративными и телекоммуникационными компаниями с более чем











LOCAWEB





































#### ЭКСПЕРТЫ



**ДЖЭЙМС ГОСЛИНГ**Создатель Java *Независимый Директор* 



МАЙКЛ "MONTY" ВИДЕНИУС
Основатель MySQL и MariaDB
Технологический советник



МАРК ЗБИКОВСКИ

Бывший архитектор Microsoft

Технологический советник



**РАСМУС ЛЕРДОРФ**Создатель РНР *Технологический советник* 



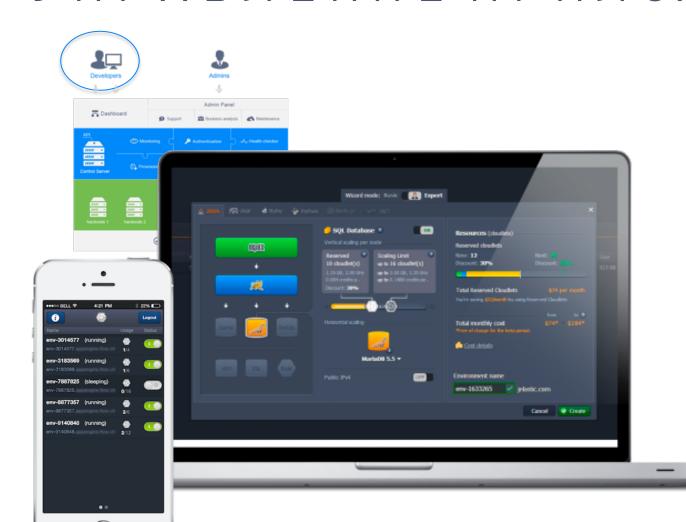
**ИГОРЬ СЫСОЕВ**Основатель NGINX
Технологический советник



ДЭВИД БЛЕВИНС
Основатель Apache TomEE, OpenEJB и
Geronimo
Технологический советник



#### УПРАВЛЕНИЕ ПРИЛОЖЕНИЯМИ



Панель управления разработчика для создания и управления окружениями

- сервера приложений
- балансировка нагрузки
- кластеризация и высокодоступность
- Сохраняет настройки серверов в окружении в актуальном состояии
- Определяет параметры для автомасштабирования приложений
- Обеспечивает контроль разворачивания приложений, применения патчей, обновлений и откатов



#### ДОСТУПНОСТЬ ПРИЛОЖЕНИЙ







Jelastic поддерживает любое стандартное приложение без изменений кода. Отсутствие проприетарных и сложных API.

JDK PHP Ruby Python Node.js

.NET (Q1/2015)

Tomcat TomEE Jetty Glassfish

JBoss Apache NGINX MySQL MariaDB PostgreSQL

Mongo DB Couch DB

Neo4j

Cassandra

Redis

Maven Build Node Memcached

Cartridges

Docker (Q4)

Clojure

jRuby

Coldfusion

Groovy

Scala

GIT

SVN

Eclipse

IDEA

**NetBeans** 

Maven plugin

Ant plugin





#### ПОДДЕРЖКА РАСШИРЕНИЙ

Jelastic поддерживает спецификацию OpenShift Cartridges, что позволяет использовать дополнительные стэки, созданные сообществом OpenShift:

- Балансировщики нагрузки
- Сервера приложений
- Сервера БД
- Кэширующие сервера, и т.д.





программные стэки, в том числе проприетарные.

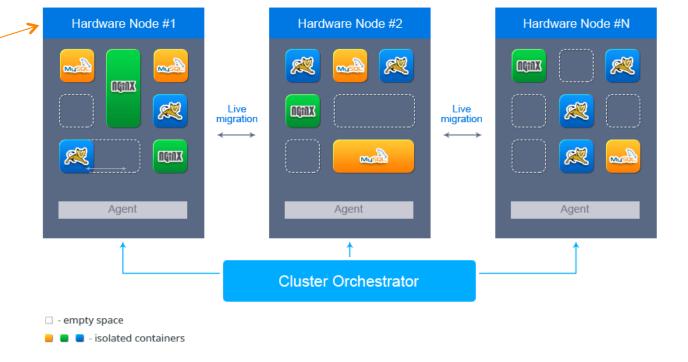




#### АРХИТЕКТУРА РАЗВЕРТЫВАНИЯ







Platform-as-Infrastructure = DevOps Platform



#### ВЕРТИКАЛЬНОЕ МАСШТАБИРОВАНИЕ В ОБЛАКЕ

Виртуальные машины хорошо масштабируются по HDD, Network, IO

#### Ho:

- Гипервизор резервирует память, процессорную мощность под виртуальные машины на этапе создания
- Невозможно изменить выделенную память/СРU во время работы ВМ: необходима перезагрузка
- «Масштабирование вниз» труднодостижимо



## А В Т О М А Т И Ч Е С К О Е М А С Ш Т А Б И Р О В А Н И Е

- Jelastic предоставляет эффективный способ добавлять и удалять ресурсы толь Ках Ошт Калы Дый да это необходимо:

   Автоматическое вертикальное масштабирование
   Автоматическое горизонталы но масштабирование
   Быстрое расширение облачного к Reserved Resources
- Во время пиковых нагрузок Jelastic добавляет больше ресурсов ВМ и забирает их обратно, когда они становятся не нужны
- Jelastic обеспечивает эластичность облака за счет автоматического горизонтального масштабирования (вверх и вниз) на основе
   показателей нагрузки



#### КОНТЕЙНЕРНАЯ ВИРТУАЛИЗАЦИЯ

- Parallels Virtuozzo Containers виртуализация на уровне ОС
- Самые низкие накладные расходы на виртуализацию
- Возможность автоматического вертикального масштабирования по RAM, CPU, HDD, Network, IO
- Пользователь не платит за потребление памяти и процессора ядра ВМ
- Дедупликация файловой системы
- DRS





#### JELASTIC VS. AMAZON

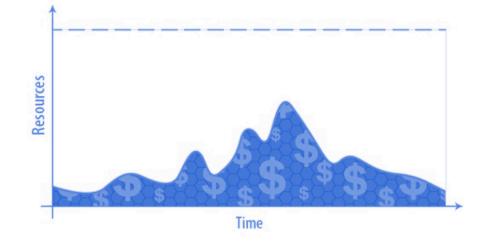
#### **Amazon EC2**

- Предлагает только сервера с фиксированными лимитами потребления ресурсов
- Гранулярность 2X
- Оплата за лимиты
- Единичный пик нагрузки приводит к масштабированию до следующего лимита ресурсов

# SSOURCE STATE OF TIME

#### Jelastic Platform-as-Infrastructure

- Грануляция ресурсов 200МГц СРU,
   128Мб RAM
- Оплата только за фактическое потребление
- Автоматическое вертикальное и горизонтальное





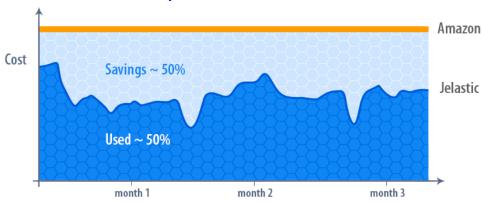


#### ЭКОНОМИЯ РЕАЛЬНЫХ ПРОЕКТОВ

http://onde.ir: travel



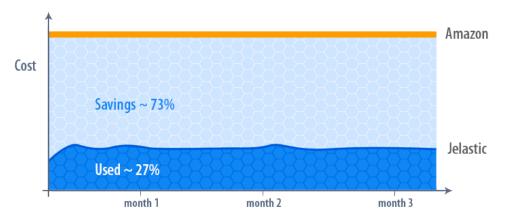
http://iidf.ru: financial



#### http://murastudio.com: gaming



#### http://saprigrat.whelastic.net: academic







#### ОСОБЕННОСТЬ МАСШТАБИРОВАНИЯ JVM

- Физическое потребление памяти JVM может только увеличиваться
- Освободившаяся после работы GC память не отдается обратно в операционную систему
- Ее нельзя перераспределять между другими приложениями в облаке
- Разные **GC** по разному работают с большим объемом памяти



#### ВЕРТИКАЛЬНОЕ MACШТАБИРОВАНИЕ В JELASTIC

- Результат 6 месяцев исследований
- Обеспечивается автоматическое вертикальное масштабирования вверх и вниз для Java приложений
- Работает в GI и Serial GC
- Полный эффект достигается за счет применения контейнерной виртуализации
- Абсолютно прозрачно и выгодно для пользователя

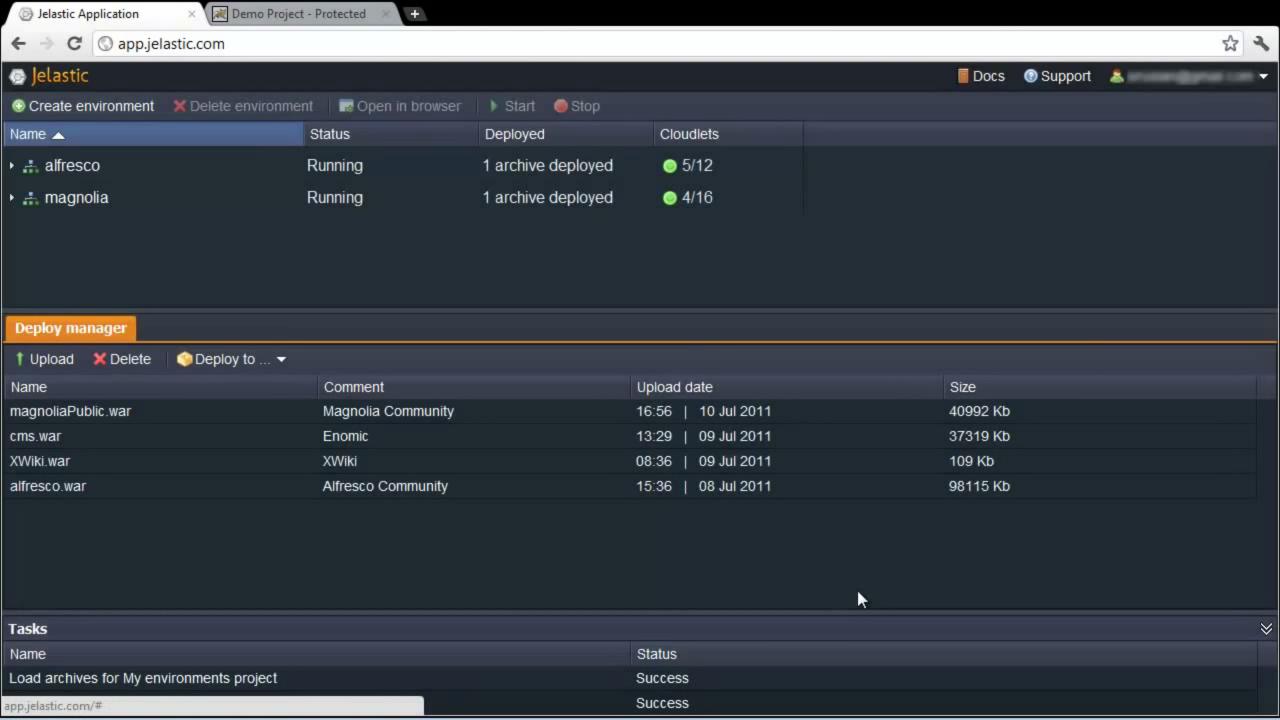




#### ТЕСТОВОЕ ПРИЛОЖЕНИЕ

```
<html>
<body>
<h1>Hello!</h1>
<%
   byte[] data = (byte[]) session.getAttribute("test-data");
    if (data == null) data = new byte[1024 * 1024 * 10];
    request.getSession().setAttribute("test-data", data);
%>
Your session ID <%=session.getId()%>
Your session data size <%=data.length%> bytes
</body>
</html>
```





#### МАГИЧЕСКИЕ ПАРАМЕТРЫ

- -Xminf0.1 параметр определяет минимальное свободное пространство в heap и поручает JVM расширить его, если после выполнения сборки мусора нет, по крайней мере, 10% свободного пространства
- -Xmaxf0.3 параметр определяет насколько heap расширен и поручает JVM уплотнить его, если объем свободного пространства превышает 30%
- -XX:-UseAdaptiveSizePolicy





## АРХИТЕКТУРА АВТОМАТИЧЕСКОГО МАСШТАБИРОВАНИЯ JVM

- 1. В зависимости от Xmx выбирается подходящий GC
  - -XX:+UseParNewGC
  - -XX:+UseG1GC
- 2. Jelastic GC Java agent внедряется в каждую JVM в момент запуска
- 3. Jelastic GC Java agent периодически определяет необходимость освобождения памяти
- 4. Jelastic GC Java agent инициализирует Full GC для компактинации
- 5. Если у JVM нет лишней свободной памяти Full GC не вызывается





#### ЧЕМ МОЖНО УПРАВЛЯТЬ?

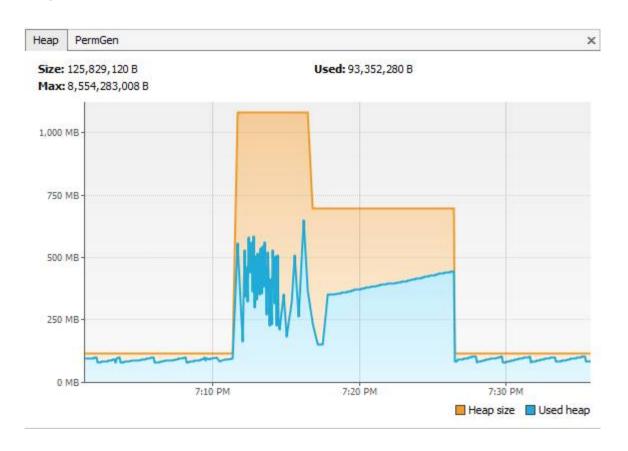
- Вкл/Выкл
- Частота проверки необходимости масштабирования вниз
- Xminf
- Xmaxf
- Xmn





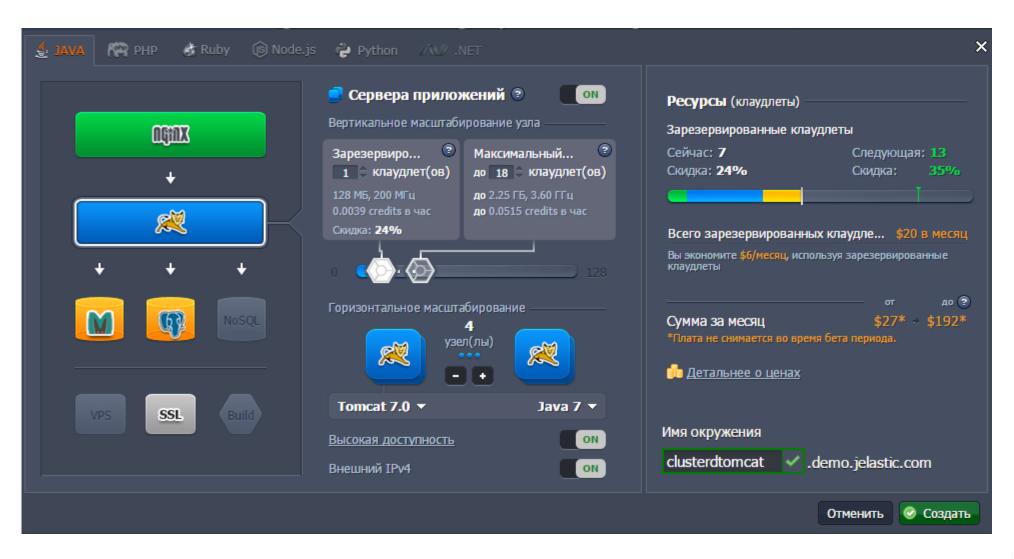
#### ГРАФИК VISUAL VM НА РЕАЛЬНОМ ПРОЕКТЕ

"I'm loving the new GC Agent! Look at the money it is saving!" — Katherine Morgan Demchinsky





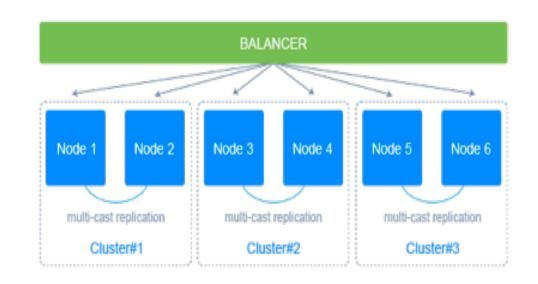
#### ГОРИЗОНТАЛЬНОЕ МАСШТАБИРОВАНИЕ





#### STATEFUL ПРИЛОЖЕНИЯ

- Репликация частей файловой системы
- Репликация сессий через мультикаст
- «Sticky-балансировка»
- Высокая доступность и Anti-affinity «умное» распределение между физическими





ерами

## АВТОМАТИЧЕСКОЕ ГОРИЗОНТАЛЬНОЕ МАСШТАБИРОВАНИЕ





#### ОТКАЗОУСТОЙЧИВОСТЬ





## Бесплатный пробный период::

www.jelastic.com/connect/free-trial/

Дополнительная информация:

