Опыт использования Chef в высоконагруженных проектах

Докладчик : Дмитрий Лавриненко

Luxoft

Front-End

- *Nginx*
- *AWS CloudFront*
- *S3*

Back-End

- *NodeJS*
- *PHP-FPM*

Data Storage

- *MongoDB*
- *Elasticsearch*
- *Redis*
- *Memcached*

Методы администрирования

- *Manual*
- *Scripts*
- *Configuration Management (CM)*

Что позволяет делать Chef?

- *Настройка AMI на этапе запуска*
- *Поиск по ролям внутри Chef*
- *Удобный доступ к фермам серверов*
- *Гибкость настройки конфигураций*

Интеграция с AWS

- *CloudFormation*
- *Route53 DNS*
- *EC2*
- *etc*

Процесс масштабирования

- *Создание севера*
- *Применение роли в Chef*
- *Сервер в строю*

- *Инициация выключения/удаления сервера из Chef*
- *Вывод сервера из всех конфигураций*
- *Выключение сервера*

Шардинг баз данных

- *MongoDB*
- *ElasticSearch*
- *Redis & Memcached*

Code Deploy

- *S3 & NFS*
- *Code Versions*
- *Autodeploying*

Резервное копирование

- *AMI*
- *S3*
- *Долгосрочное хранение в Glacier*

P.S.

- *В каких случаях эффективен Chef?*
- *Всегда ли нужно создавать сложные CM архитектуры?*
- *Почему так важно хранить конфигурации в CM системах*

Feedback

Dmitry Lavrinenko

dlavrinenko@luxoft.com
nonflet@gmail.com