

DevOps Marathon

Длительность курса: 9 академических часов

1 DevOps для руководителей

1 **Эволюция DevOps на примере небольшой компании. Алик Курдюков, СТО UnitedTraders**

в докладе рассказывается, как развивалась инфраструктура компании. Все начиналось с одного разработчика и все ставили руками. С ростом количества установок перешли на скрипты. В дальнейшем количество сервисов увеличивалось, и со второй попытки внедрили infrastructure as a code на ansible. В докладе рассказывается, как компания ушла из облаков, а потом пришла к частному облаку. И самая интересная часть доклада - объясняется почему делался каждый шаг, как это экономически обосновывалось. В заключении освещается, как будет происходить дальнейшая эволюция инфраструктуры с использованием OpenShift.

2 **Наследование устаревших систем и процессов или первые 90 дней в роли СТО. Leon Fayer, СТО Teaching Strategies**

реальная история, в картинках, о наследование устаревших систем и ожидаемых (и неожиданных) проблем которые часто идут в комплекте. История о приоритете задач, систематизированном подходе, внедрение Agile, и работе с людьми. И адаптации к ситуациям когда весь прошлый опыт оказывается бесполезным.

Рассказано, какие проблемы (технические, культурные, процессные и даже, как ни странно, юридические) встречаются на пути DevOps-трансформации.

В докладе рассказано, какие шаги стоит предпринять в первую очередь, чтобы достичь быстрых побед, а на чем надо сосредоточиться в долгосрочном плане, потому что быстрых результатов там не получить, как подружить бизнес и разработку, как изменить орг структуру, как начать прививать новую культуру и какие кадровые решения необходимо предпринять.

3 **Почему вы умрете без DevOps. Николай Кныш, СТО Райффайзенбанк**

за последние три-четыре года многие компании решили перевести продуктивное развитие на рельсы аджайл. Все понимают выгоду от общения со своими клиентами, частых релизов, делегирования решений задач в команды. При этом, мало кто понимает, что такая трансформация означает для ИТ дирекции. Даже если вам удастся запустить пару скрам команд, масштабирование трансформации обязательно споткнется об инженерную культуру, о проблемы с совместным владением кода и о снижение стабильности. Любого из этих пунктов достаточно, чтобы убить трансформацию. В докладе рассказано о том, как перестроить ИТ организационно, чтобы можно было централизованно отвечать на эти вызовы.

1 Спасение из джунглей: единый конвейер доставки ПО. Anton Weiss, Otomato Software

что делать когда вы строили, строили, и наконец построили — и теперь у вас десятки приносящих доход служб, основанных на огромном количестве технологии? Как продолжить поддерживать быстрые темпы изменений, не разочаровывая пользователей, но при этом добавить этому всему управляемость и прозрачность?

В докладе рассказывается о том, как достичь разумного баланса между скоростью изменения и порядком. Предлагается ввести некоторый стандарт - Golden Path (Золотой Путь), который описывает, как различные компоненты собираются, тестируются, выкатываются, мониторятся и т.д. При этом этот стандарт не является “железным”, что позволяет сохранить гибкость. В докладе приводится один из возможных наборов технологий для такого стандарта из опыта консультирования компании Otomato: Jenkins Pipeline, Helm Charts, Kubernetes, Hashicorp Vault и т.д.

Также подробно описывается, как сделать универсальный Pipeline в Jenkins, из каких частей он должен состоять, чтобы его удобно было переиспользовать разным командам.

2 **Security and K8s.**
Сергей Бердников,
Золотая корона

всегда есть боязнь нового и с первого взгляда может показаться что риски с точки зрения ИБ при использовании таких технологий как k8s возрастают, на самом деле это не совсем так. С применением новых технологий должны так же меняться и подходы с точки зрения ИБ, и это действительно сложно и дорого, так как безопасность очень не инертная среда и все выработанные правила и регламенты написаны кровью.

При правильном подходе риски становятся меньше, но это не дается бесплатно, над этим нужно плотно работать. В докладе рассказано, какие сдвиги в мышление должны произойти в связи с внедрением k8s и сделать обзор на какие инструменты стоит обратить внимание.

3 **Современный Паппет, как с ним жить и зачем он нужен. Андрей Кононов, независимый консультант**

этот доклад будет про то как жить с современным Puppet и почему стоит попробовать на него обновиться. В докладе приведен краткий обзор фундаментальных изменений в версии 6.x (типы, оркестрация с Puppet bolt, Puppet CA, ERP шаблоны, интеграция с PuppetDB и т.д.). Рассказано, что нового появилось в экосистеме Puppet (rspec-puppet, Hiera, Puppet Forge и т.д.). Отдельно отмечается, как правильно организовывать репозиторий с Puppet кодом. Далее докладчик рассказывает о том, какие компоненты есть в инфраструктуре Puppet и как их правильно мониторить. Также в докладе дано несколько примеров использования разных подходов к устройству жизни с Puppet, и, конечно, несколько примеров того как делать не надо.

4 **Реальные примеры использования AWS Lambda для инфраструктурных задач. Артем Никитин, Software Engineer, Elastic**

в этом докладе рассказывается, что такое Serverless-подход, какие преимущества он дает и какие у него перспективы в ближайшем будущем. Далее в докладе раскрывается, что такое AWS Lambda, как одна из самых популярных реализаций Serverless-подхода, о ценовой модели и о преимуществах, которые она дает, по сравнению с традиционным подходом, таких как автомасштабирование, отказоустойчивость, безопасность, балансировка нагрузки, мониторинга и т.д.

Также в докладе рассмотрены кейсы из реального опыта использования Lambda. Далее рассказано о том, как разрабатывать на Lambda с реальными примерами кода, показывается, как тестировать и выкатывать приложения Lambda, а также рассказано об основных проблемах, которые при этом встречаются.