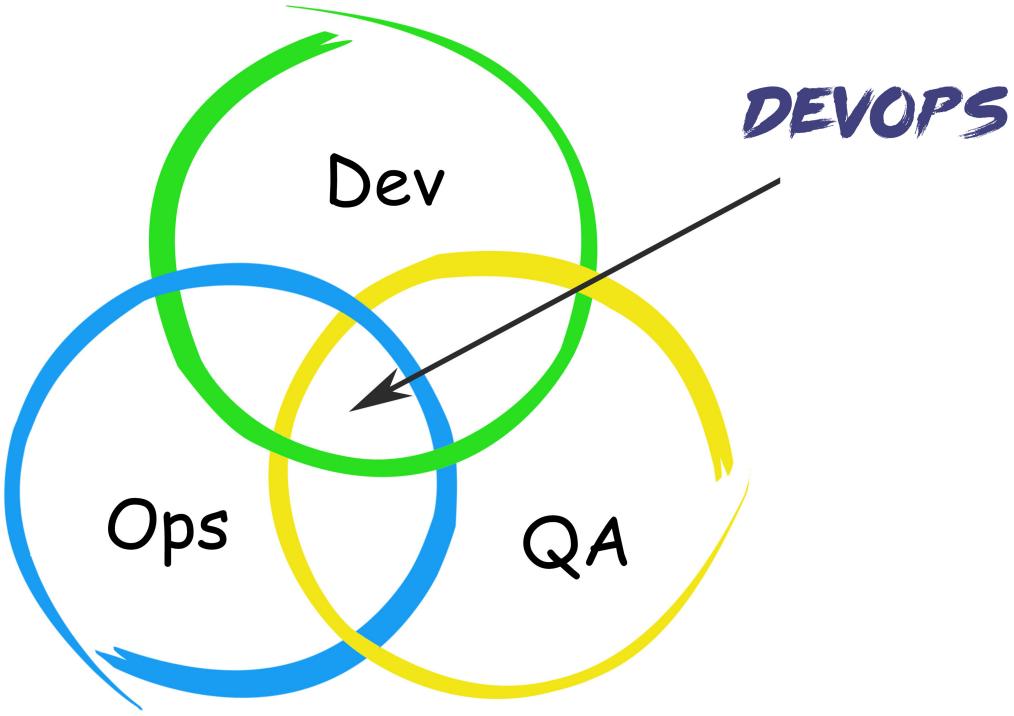
给 DevOps 初学者的入门指南

2016年09月23日
当我们谈到 DevOps 时，可能讨论的是：流程和管理，运维和自动化，架构和服务，以及文化和组织等等概念。那么，到底什么是"DevOps"呢？

什么是DevOps

随着软件发布迭代的频率越来越高，传统的「瀑布型」（开发—测试—发布）模式已经不能满足快速交付的需求。2009 年左右 DevOps 应运而生，简单地来说，就是更好的优化开发(DEV)、测试(QA)、运维(OPS)的流程，开发运维一体化，通过高度自动化工具与流程来使得软件构建、测试、发布更加快捷、频繁和可靠。



关于 DevOps 是什么，DevOps 的合著者 John Willis 写了一个非常好的帖子，在这里.

Devops 的好处与价值

在2016 DevOps 新趋势调查报告显示，74% 的公司在尝试接受 DevOps，那么 Devops 有哪些好处与价值呢？

- 代码的提交直接触发：消除等待时间，快速反馈

- 每个变化对应一个交付管道：使问题定位和调试变得简单
- 全开发流程高效自动化：稳定，快速，交付结果可预测
- 持续进行自动化回归测试：提升交付质量
- 设施共享并按需提供：资源利用最大化

以上可以看出，DevOps 的好处更多基于在于持续部署与交付，这是对于业务与产品而言。而 DevOps 始于接受 DevOps 文化与技术方法论，它是部门间沟通协作的一组流程和方法，有助于改善公司组织文化、提高员工的参与感。

Devops与持续集成

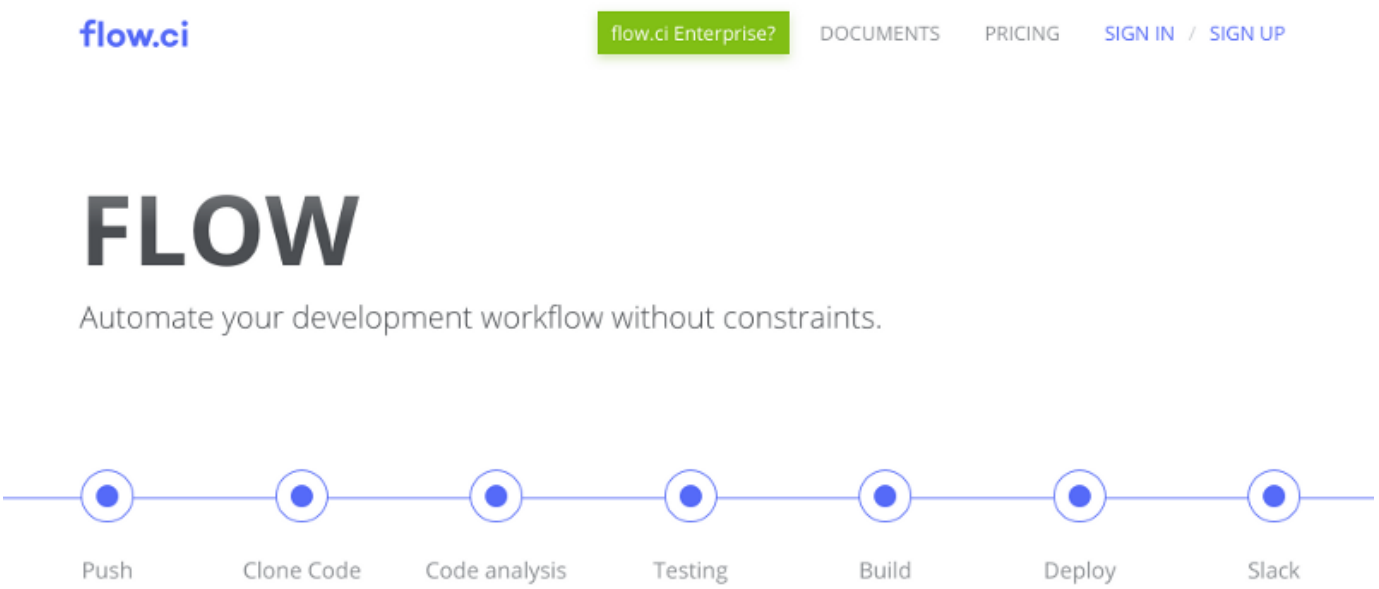
DevOps 是一个完整的面向IT运维的工作流，以 IT 自动化以及持续集成（CI）、持续部署（CD）为基础，来优化程式开发、测试、系统运维等所有环节。

纵观各个 DevOps 实践公司的技术资料，最全面最经典的是 flickr 的10+ deploys per day最佳实践提到的 DevOps Tools 的技术关键点:

```
1.Automated infrastructure（自动化，系统之间的集成）
2.shared version control（SVN共享源码）
3.one step build and deploy（持续构建和部署）
4.feature flags（主干开发）
5.Shared metrics
6.IRC and IM robots（信息整合）
```

以上的技术要点由持续集成/部署一线贯穿，主干开发是进行持续集成的前提，自动化以及代码周边集中管理是实施持续集成的必要条件。毫无疑问，DevOps 是持续集成思想的延伸，持续集成/部署是 DevOps 的技术核心，在没有自动化测试、持续集成/部署之下，DevOps就是空中楼阁。

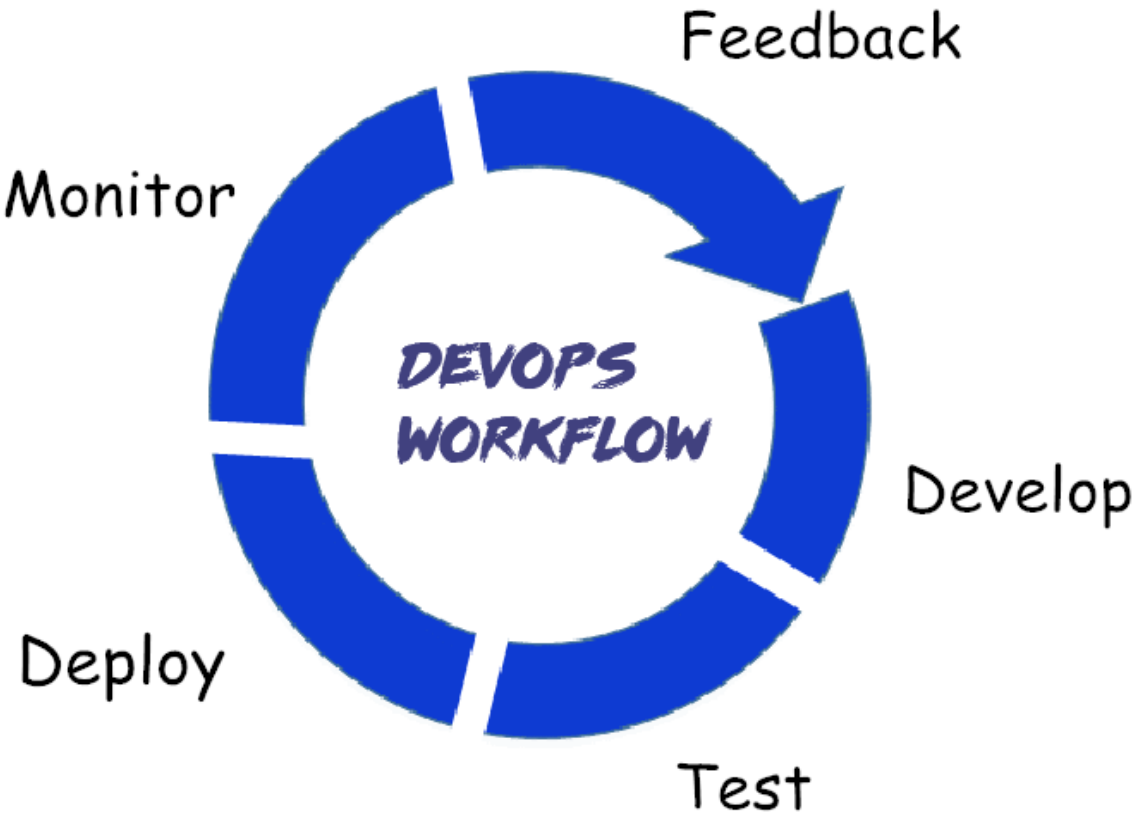
我们做了一款 Hosted 持续集成产品—— flow.ci ，它融入了 workflow 机制的持续集成（CI）服务，也可以理解为自动化流程平台，除了集成代码、编译、测试之外，还可以集成常用的工具、灵活自定义流程，帮助你们塑造一个更优秀智能的 DevOps 环境。



DevOps 的技术栈与工具链

Everything is Code，DevOps 也同样要通过技术工具链完成持续集成、持续交付、用户反馈和系统优化的整合。Elasticbox 整理了 60+ 开源工具与分类，其中包括版本控制&协作开发工具、自动化构建和测试工具、持续集成&交付工具、部署工具、维护工具、监控，警告&分析工具等等，
补充了一些国内的服务，可以让你更好的执行实施 DevOps 工作流。

- **版本控制&协作开发**：GitHub、GitLab、BitBucket、SubVersion、Coding、Bazaar
- **自动化构建和测试**:Apache Ant、Maven 、Selenium、PyUnit、QUnit、JMeter、Gradle、PHPUnit
- **持续集成&交付**:Jenkins、Capistrano、BuildBot、Fabric、Tinderbox、Travis CI、flow.ci、Continuum、LuntBuild、CruiseControl、Integrity、Gump、Go
- **容器平台**: Docker、Rocket、Ubuntu（LXC）、第三方厂商如（AWS/阿里云）
- **配置管理**：Chef、Puppet、CFengine、Bash、Rudder、Powershell、RunDeck、Saltstack、Ansible
- **微服务平台**：OpenShift、Cloud Foundry、Kubernetes、Mesosphere
- **服务开通**：Puppet、Docker Swarm、Vagrant、Powershell、OpenStack Heat
- **日志管理**：Logstash、CollectD、StatsD
- **监控，警告&分析**：Nagios、Ganglia、Sensu、zabbix、ICINGA、Graphite、Kibana



顺便再分享一个 DevOps BookMarks，涉及了DevOps方方面面的工具和内容，有兴趣的同学可以去学习下。

DevOps 最佳实践

自 2009 年提出 DevOps 的概念起，很多公司都开始实施 DevOps，国外比较著名的有Amazon 、Google、Facebook 等，国内著名的有百度、华为、阿里等。Amazon 是 DevOps 最佳实践的最有说服力的代表之一。这是 Amazon 在 Why We Need DevOps 一个月的 DevOps 快照：

> 11.6 seconds: 平均部署时长 (工作日)
> 1,079: 一小时的最大部署量
> 10,000: 主机平均并发接收部署量
> 30,000: 主机最高并发接收部署量

从早期的大型 SOA (Service Oriented Architecture)到 DevOps 文化的形成，Amazon 的每个工程师都可以完全独立地编写代码，测试代码，版本管理，部署上线，服务监测等任务。这套内部强大的 DevOps 文化最终形成核聚变， Amazon 一跃成为世界级别的云服务领导者 —— Amazon Web Services (AWS)。

除了 Amazon 外还有一些国内外的 DevOps 实践公司，一起来看看。

- **flickr**

最全面最经典的是 flickr 的10+ deploys per day，简直是 DevOps 教科书般的存在。

- **百度**

百度技术团队是如何利用 DevOps，来看看解密百度持续交付方法与实践。

- **Netflix**

解密Netflix 技术团队在整个 DevOps 过程中使用的部署工具和服务.

- **Etsy**

2009年，Etsy建立自己的工具来更好更快地部署发布，「Etsy 如何应用 DevOps」值得一读。

- **LinkedIn**

2009年，LinkedIn 团队就开始使用自动化部署工具，用于管理在1000+节点环境下发布上千个应用/服务的复杂性。这是 LinkedIn 自己造的轮子 >>Deployment and Monitoring Automation with glu.

- **Airbnb**

Airbnb 作为第三方平台公司，需要迅速发布多个小型部署。关于 Airbnb 的数据和基础设施，可以参考这个slides。

- **Starbucks**

星巴克的 DevOps 计划>> Starbucks Announces #DevOpsTogether.

- **Ancestry.com**

Ancestry.com 是 DevOps 运动的早期采用者，是 Continuous Delivery 和 DevOps 运动的先锋。想了解更多关于他们的过程、迁移和 DevOps 文化，不妨查看一下他们的系列文章 http://blogs.ancestry.com/techroots/category/devops/。

DevOps = Culture + Tools

如果想整个业务部署 DevOps，不但需要软性要求即从上而下的培养 DevOps 文化自上而下地进行探索，也有硬性工具链要求，才能获得更高质量的软件交付。

最后，不论你是技术Leader，还是一名Dev、QA 或 Ops，实现全面的 DevOps 非常理想化也十分有挑战，希望这份 DevOps 初学者指南是一个好的开始:)