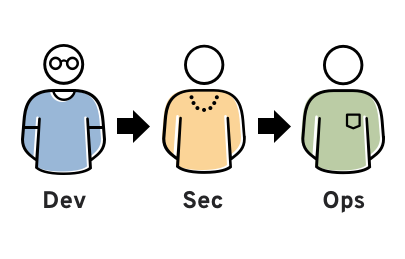
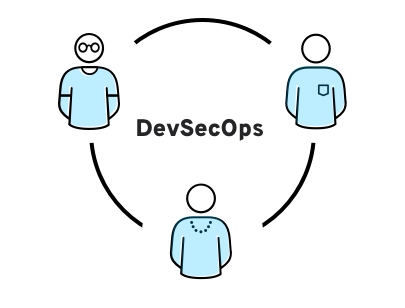
**什么是Devsecops**

我们知道，DevOps 不仅仅涉及开发和运维团队。而如果您想充分发挥出 DevOps 的敏捷性和响应力，就必须将[IT安全防护](https://www.redhat.com/zh/topics/security)融入应用的整个生命周期中 。

为什么呢？从前，安全防护只是特定团队的责任，在开发的最后阶段才会介入。当开发周期长达数月、甚至数年时，这样做没什么问题；但是现在，这种做法现在已经行不通了。采用 DevOps 可以有效推进快速频繁的开发周期（有时全程只有数周或数天），但是过时的安全措施则可能会拖累整个流程，即使最高效的 DevOps 计划也可能会放慢速度。



如今，在 DevOps 协作框架下，安全防护是整个 IT 团队的共同责任，需要[贯穿](https://www.redhat.com/zh/solutions/agile-integration)至整个生命周期的每一个环节。这个理念非常重要，因此催生出了“DevSecOps”一词，强调必须为 DevOps 计划打下扎实的安全基础。



DevSecOps 意味着从一开始就要考虑应用和基础架构的安全性；同时还要让某些安全网关实现[自动化](https://www.redhat.com/zh/topics/automation)，以防止 DevOps 工作流程变慢。选择合适的工具来持续集成安全防护（比如在[集成开发环境（IDE）](https://www.redhat.com/zh/topics/middleware/what-is-ide)中集成安全防护功能）有助于实现这些目标。但是高效的 DevOps 安防需要的不仅是新工具。它更需要整个公司实现 DevOps 文化变革，从而尽早集成安全团队的工作。

**DevOps 一体化安全防护**

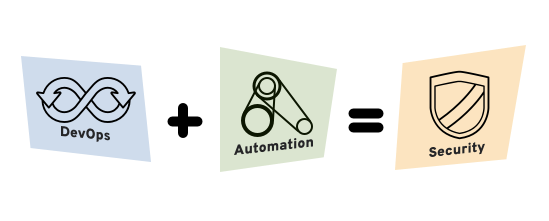
无论您习惯叫它“DevOps”还是“DevSecOps”，最好确保安全防护融入了软件的整个生命周期中。DevSecOps 就是要内置安全防护，而不是仅仅在应用和数据层面做文章。如果把安全问题留到开发流程的最后再考虑，那么企业即便是采用了 DevOps 方法，也会重回冗长开发周期的老路，而这就是大家从一开始就想要避免的情况。

DevSecOps 强调，在 DevOps 计划刚启动时就要邀请安全团队来确保信息的安全性，并制定自动安全防护计划，[实现持续 IT 防护](https://www.redhat.com/zh/technologies/guide/it-security)。它还强调，要帮助开发人员从代码层面确保安全性；在这个过程中，安全团队需要针对已知的威胁共享掌握的全局信息、提供反馈并进行智能分析。由于 DevSecOps 并非始终着眼于较为传统的应用开发模式，所以这可能还包括为开发人员提供新的安全培训。

那么，怎样才算是真正地实现了安全防护一体化？对于新手而言，优良的 DevSecOps 策略应能确定风险承受能力并进行风险/收益分析。在一个特定的应用中，需要配备多少个安全控制功能？对于不同的应用，上市速度又有多重要？由于在管道中运行手动安全检查可能会非常耗时，所以自动执行重复任务是 DevSecOps 的关键所在。

## ****DevOps 安全防护自动化****

要推行 DevOps，企业需要：确保采用时间短、频率高的开发周期；采取安全措施，最大限度地缩短运维中断时间；及时应用创新技术，如[容器](https://www.redhat.com/zh/topics/containers)和[微服务](https://www.redhat.com/zh/topics/microservices)；同时，还要促使以往各自为阵的团队加强合作——这一切对所有企业来说都是艰巨的任务。上述所有措施都是以人员为根本，都需要企业内部协同合作；但是，要想在 DevSecOps 框架中实现这些人员的变革，[自动化](https://www.redhat.com/zh/topics/automation/whats-it-automation)是关键所在（[点击查看自动化的带来的优势](https://www.redhat.com/zh/topics/automation)）。



那么，企业应该在哪些方面实现自动化？具体又该怎么做呢？这份的[企业自动化检查清单](https://www.redhat.com/zh/resources/enterprise-automation-checklist?source=searchresultlisting)也许能够为您提供答案。企业应该退后一步，以全局视角，着眼整个开发和运维环境。其中涉及：源控制存储库；容器注册表；持续集成和持续部署（CI/CD）管道；应用编程接口（API）的管理、编排和发布自动化；以及运维管理和监控。

全新的自动化技术已帮助企业应用了更敏捷的开发实践，也在促进新的安全措施方面起到了重要作用。但是，自动化并不是近年来 IT 领域发生的唯一变化。现在，对于大多数 DevOps 计划而言，容器和微服务等[云原生](https://www.redhat.com/zh/topics/cloud-native-apps)技术也是一个非常重要的组成部分。所以，企业必须及时调整 DevOps 安全措施。

## ****针对容器和微服务的 DevOps 安全防护****

容器可以实现更大的规模和更灵活的基础架构，从而改变了许多组织开展业务的方式。因此，DevOps 的安全防护实践必须适应新环境并遵循[容器特有的安全准则](https://csrc.nist.gov/publications/detail/nistir/8176/final)。云原生技术不适合用来落实静态安全策略和检查清单。相反，企业必须确保应用和基础架构生命周期的每个阶段都有集成其中的持续安全防护。

DevSecOps 意味着，安全防护要贯穿应用开发的整个过程。要实现集成进管道，我们既需要一种全新的思维方式，也需要合适的新工具。因此，DevOps 团队应该实现安全防护自动化，从而保护整体环境和数据，同时实现持续集成/持续交付流程——可能还要确保容器中的微服务的安全。



