

DevOps & CI/CD 2022 年最新常见面试题

汇总

目录

DevOps & CI/CD 常见面试题汇总.....	1
DevOps 术语和定义.....	2
实施 DevOps 的原因.....	4
如何有效实施 DevOps	6
如何有效实施 CI/CD.....	9
每种术语之间的差异.....	12

DevOps 术语和定义

1. 什么是 DevOps

答：用最简单的术语来说，DevOps 是产品开发过程中开发（Dev）和运营（Ops）团队之间的灰色区域。DevOps 是一种在产品开发周期中强调沟通，集成和协作的文化。因此，它消除了软件开发团队和运营团队之间的孤岛，使他们能够快速，连续地集成和部署产品。

2. 什么是持续集成

答：持续集成（Continuous integration，缩写为 CI）是一种软件开发实践，团队开发成员经常集成他们的工作。利用自动测试来验证并断言其代码不会与现有代码库产生冲突。理想情况下，代码更改应该每天在 CI 工具的帮助下，在每次提交时进行自动化构建（包括编译，发布，自动化测试），从而尽早地发现集成错误，以确保合并的代码没有破坏主分支。

3. 什么是持续交付

答：持续交付（Continuous delivery，缩写为 CD）以及持续集成为交付代码包提供了完整的流程。在此阶段，将使用自动构建工具来编译工件，并使其准备好交付给最终用户。它的目标在于让软件的构建、测试与发布变得更快以及更频繁。这种方式可以减少软件开发的成本与时间，减少风险。

4. 什么是持续部署

答：持续部署（Continuous deployment）通过集成新的代码更改并将其自动交付到发布分支，从而将持续交付提升到一个新的水平。更具体地说，一旦更新通过了生产流程的所有阶段，便将它们直接部署到最终用户，而无需人工干预。因此，要成功利用连续部署，软件工件必须先经过严格建立的自动化测试和工具，然后才能部署到生产环境中。

5. 什么是持续测试及其好处

答：连续测试是一种在软件交付管道中尽早、逐步和适当地应用自动化测试的实践。在典型的 CI/CD 工作流程中，将小批量发布构建。因此，为每个交付手动执行测试用例是不切实际的。自动化的连续测试消除了手动步骤，并将其转变为自动化例程，从而减少了人工。因此，对于 DevOps 文化而言，自动连续测试至关重要。

持续测试的好处：

- b. 确保构建的质量和速度。
- c. 支持更快的软件交付和持续的反馈机制。
- d. 一旦系统中出现错误，请立即检测。
- e. 降低业务风险。在潜在问题变成实际问题之前进行评估。

6. 什么是版本控制及其用途？

答：版本控制（或源代码控制）是一个存储库，源代码中的所有更改都始终存储在这个代码仓库中。版本控件提供了代码开发的操作历史记录，追踪文件的变更内容、时间、人等信息忠实地记录下来。版本控制是持续集成和持续构建的源头。

7. 什么是 Git?

答：Git 是一个分布式版本控制系统，可跟踪代码存储库中的更改。利用 GitHub 流，Git 围绕着一个基于分支的工作流，该工作流随着团队项目的不断发展而简化了团队协作。

实施 DevOps 的原因

1. DevOps 为什么重要？DevOps 如何使团队在软件交付方面受益？

答：在当今的数字化世界中，组织必须重塑其产品部署系统，使其更强大，更灵活，以跟上竞争的步伐。

这就是 DevOps 概念出现的地方。DevOps 在为整个软件开发管道（从构思到部署，再到最终用户）产生移动性和敏捷性方面发挥着至关重要的作用。DevOps 是将不断更新和改进产品的更简化，更高效的流程整合在一起的解决方案。

2. 解释 DevOps 对开发人员有何帮助？

答：在没有 DevOps 的世界中，开发人员的工作流程将首先建立新代码，交付并集成它们，然后，操作团队有责任打包和部署代码。之后，他们将不得不等待反馈。而且如果出现问题，由于错误，他们将不得不重新执行一次，在项目中涉及的不同团队之间的无数手动沟通。

由于 CI/CD 实践已经合并并自动化了其余任务，因此应用 DevOps 可以将开发人员的任务简化为仅构建代码。随着流程变得更加透明并且所有团队成员都可以访问，将工程团队和运营团队相结合有助于建立更好的沟通和协作。

3. 为什么 DevOps 最近在软件交付方面变得越来越流行？

答：DevOps 在过去几年中受到关注，主要是因为它能够简化组织运营的开发，测试和部署流程，并将其转化为业务价值。

技术发展迅速。因此，组织必须采用一种新的工作流程-DevOps 和 Agile 方法-来简化和刺激其运营，而不能落后于其他公司。DevOps 的功能通过 Facebook 和 Netflix 的持续部署方法所取得的成功得到了清晰体现，该方法成功地促进了其增长，而没有中断正在进行的运营。

4. CI/CD 有什么好处？

答：CI 和 CD 的结合将所有代码更改统一到一个单一的存储库中，并通过自动化测试运行它们，从而在所有阶段全面开发产品，并随时准备部署。

CI/CD 使组织能够按照客户期望的那样快速，高效和自动地推出产品更新。

简而言之，精心规划和执行良好的 CI/CD 管道可加快发布速度和可靠性，同时减轻产品的代码更改和缺陷。这最终将导致更高的客户满意度。

5. 持续交付有什么好处？

答：通过手动发布代码更改，团队可以完全控制产品。在某些情况下，该产品的新版本将更有希望：具有明确业务目的的促销策略。

通过自动执行重复性和平凡的任务，IT 专业人员可以拥有更多的思考能力来专注于改进产品，而不必担心集成进度。

6. 持续部署有哪些好处？

答：通过持续部署，开发人员可以完全专注于产品，因为他们在管道中的最后任务是审查拉取请求并将其合并到分支。通过在自动测试后立即发布新功能和修复，此方法可实现快速部署并缩短部署持续时间。

客户将是评估每个版本质量的人。新版本的错误修复更易于处理，因为现在每个版本都以小批量交付。

如何有效实施 DevOps

1. 定义典型的 DevOps 工作流程，典型的 DevOps 工作流程可以简化为 4 个阶段：

- 版本控制：这是存储和管理源代码的阶段。版本控件包含代码的不同版本。
- 持续集成：在这一步中，开发人员开始构建组件，并对其进行编译，验证，然后通过代码审查，单元测试和集成测试进行测试。
- 持续交付：这是持续集成的下一个层次，其中发布和测试过程是完全自动化的。CD 确保将新版本快速，可持续地交付给最终用户。
- 持续部署：应用程序成功通过所有测试要求后，将自动部署到生产服务器上以进行发布，而无需任何人工干预。

2. DevOps 的核心操作是什么？

DevOps 在开发和基础架构方面的核心运营是 Software development:

Infrastructure:

- Provisioning
- Configuration
- Orchestration
- Deployment
- Code building
- Code coverage
- Unit testing
- Packaging
- Deployment

3. 在实施 DevOps 之前，团队需要考虑哪些预防措施？

当组织尝试应用这种新方法时，对 DevOps 做法存在一些误解，有可能导致悲惨的失败：

- DevOps 不仅仅是简单地应用新工具和/或组建新的“部门”并期望它能正常工作。实际上，DevOps 被认为是一种文化，开发团队和运营团队遵循共同的框架。
- 企业没有为其 DevOps 实践定义清晰的愿景。对开发团队和运营团队而言，应用 DevOps 计划是一项显着的变化。因此，拥有明确的路线图，

将 DevOps 集成到你的组织中的目标和期望将消除任何混乱，并从早期就提供清晰的指导方针。

- 在整个组织中应用 DevOps 做法之后，管理团队需要建立持续的学习和改进文化。系统中的故障和问题应被视为团队从错误中学习并防止这些错误再次发生的宝贵媒介。

4. SCM 团队在 DevOps 中扮演什么角色？

答：软件配置管理（SCM）是跟踪和保留开发环境记录的实践，包括在操作系统中进行的所有更改和调整。

在 DevOps 中，将 SCM 作为代码构建在基础架构即代码实践的保护下。

SCM 为开发人员简化了任务，因为他们不再需要手动管理配置过程。现在，此过程以机器可读的形式构建，并且会自动复制和标准化。

5. 质量保证（QA）团队在 DevOps 中扮演什么角色？

答：随着 DevOps 实践在创新组织中变得越来越受欢迎，QA 团队的职责和相关性在当今的自动化世界中已显示出下降的迹象。

但是，这可以被认为是神话。DevOps 的增加并不等于 QA 角色的结束。这仅意味着他们的工作环境和所需的专业知识正在发生变化。因此，他们的主要重点是专业发展以跟上这种不断变化的趋势。

在 DevOps 中，质量保证团队在确保连续交付实践的稳定性以及执行自动重复性测试无法完成的探索性测试任务方面发挥战略作用。他们在评估测试和检测最有价值的测试方面的见识仍然在缓解发布的最后步骤中的错误方面起着至关重要的作用。

6. DevOps 使用哪些工具？描述你使用任何这些工具的经验

答：在典型的 DevOps 生命周期中，有不同的工具来支持产品开发的各个阶段。因此，用于 DevOps 的最常用工具可以分为 6 个关键阶段：

持续开发：Git, SVN, Mercurial, CVS, Jira 持续整合：Jenkins, Bamboo, CircleCI

持续交付：Nexus, Archiva, Tomcat 持续部署：Puppet, Chef, Docker 持续监

控：Splunk, ELK Stack, Nagios 连续测试：Selenium, Katalon Studio

7. 如何在 DevOps 实践中进行变更管理

答：典型的变更管理方法需要与 DevOps 的现代实践适当集成。第一步是将变更集中到一个平台中，以简化变更，问题和事件管理流程。

接下来，企业应建立高透明度标准，以确保每个人都在同一页面上，并确保内部信息和沟通的准确性。

对即将到来的变更进行分层并建立可靠的策略，将有助于最大程度地降低风险并缩短变更周期。最后，组织应将自动化应用到其流程中，并与 DevOps 软件集成。

如何有效实施 CI/CD

1. CI/CD 的一些核心组件是什么？

答：稳定的 CI/CD 管道需要用作版本控制系统的存储库管理工具。这样开发人员就可以跟踪软件版本中的更改。

在版本控制系统中，开发人员还可以在项目上进行协作，在版本之间进行比较并消除他们犯的任何错误，从而减轻对所有团队成员的干扰。

连续测试和自动化测试是成功建立无缝 CI / CD 管道的两个最关键的关键。自动化测试必须集成到所有产品开发阶段（包括单元测试，集成测试和系统测试），以涵盖所有功能，例如性能，可用性，性能，负载，压力和安全性。

2. CI/CD 的一些常见做法是什么？

答：以下是建立有效的 CI / CD 管道的一些最佳实践：

- 发展 DevOps 文化
- 实施和利用持续集成
- 以相同的方式部署到每个环境
- 失败并重新启动管道
- 应用版本控制
- 将数据库包含在管道中
- 监控你的持续交付流程
- 使你的 CD 流水线流畅

3. 什么时候是实施 CI/CD 的最佳时间？

答：向 DevOps 的过渡需要彻底重塑其软件开发文化，包括工作流，组织结构和基础架构。因此，组织必须为实施 DevOps 的重大变化做好准备。

4. 有哪些常见的 CI/CD 服务器

答：Visual Studio Visual Studio 支持具有敏捷计划，源代码控制，包管理，测试和发布自动化以及持续监视的完整开发的 DevOps 系统。

TeamCity TeamCity 是一款智能 CI 服务器，可提供框架支持和代码覆盖，而无需安装任何额外的插件，也无需模块来构建脚本。

Jenkins 它是一个独立的 CI 服务器，通过共享管道和错误跟踪功能支持开发和运营团队之间的协作。它也可以与数百个仪表板插件结合使用。

GitLab GitLab 的用户可以自定义平台，以进行有效的持续集成和部署。GitLab 帮助 CI / CD 团队加快代码交付，错误识别和恢复程序的速度。

Bamboo Bamboo 是用于产品发布管理自动化的连续集成服务器。Bamboo 跟踪所有工具上的所有部署，并实时传达错误。

5. 描述持续集成的有效工作流程

答：实施持续集成的成功工作流程包括以下实践：

- 实施和维护项目源代码的存储库
- 自动化构建和集成
- 使构建自检
- 每天将更改提交到基准
- 构建所有添加到基准的提交
- 保持快速构建
- 在生产环境的克隆中运行测试
- 轻松获取最新交付物
- 使构建结果易于所有人监视

- 自动化部署

每种术语之间的差异

1. 敏捷和 DevOps 之间有哪些主要区别？

答：基本上，DevOps 和敏捷是相互补充的。敏捷更加关注开发新软件和以更有效的方式管理复杂过程的价值和原则。同时，DevOps 旨在增强由开发人员和运营团队组成的不同团队之间的沟通，集成和协作。

它需要采用敏捷方法和 DevOps 方法来形成无缝工作的产品开发生命周期：敏捷原理有助于塑造和引导正确的开发方向，而 DevOps 利用这些工具来确保将产品完全交付给客户。

2. 持续集成，持续交付和持续部署之间有什么区别？

答：持续集成（CI）是一种将代码版本连续集成到共享存储库中的实践。这种做法可确保自动测试新代码，并能快速检测和修复错误。

持续交付使 CI 进一步迈出了一步，确保集成后，随时可以在一个按钮内就可以释放代码库。因此，CI 可以视为持续交付的先决条件，这是 CI / CD 管道的另一个重要组成部分。

对于连续部署，不需要任何手动步骤。这些代码通过测试后，便会自动推送到生产环境。

所有这三个组件：持续集成，持续交付和持续部署是实施 DevOps 的重要阶段。

一方面，连续交付更适用于活跃用户已经存在的应用程序，这样事情就可以变慢一些并进行更好的调整。另一方面，如果你打算发布一个全新的软件并且将整个过程指定为完全自动化的，则连续部署是你产品的更合适选择。

3. 连续交付和连续部署之间有哪些根本区别？

答：在连续交付的情况下，主分支中的代码始终可以手动部署。通过这种做法，开发团队可以决定何时发布新的更改或功能，以最大程度地使组织受益。

同时，连续部署将在测试阶段之后立即将代码中的所有更新和修补程序自动部署到生产环境中，而无需任何人工干预。

4. 持续集成和持续交付之间的区别是什么？

答：持续集成有助于确保软件组件紧密协作。整合应该经常进行；最好每小时或每天一次。持续集成有助于提高代码提交的频率，并降低连接多个开发人员的代码的复杂性。最终，此过程减少了不兼容代码和冗余工作的机会。

持续交付是 CI / CD 流程中的下一步。由于代码不断集成到共享存储库中，因此可以持续测试该代码。在等待代码完成之前，没有间隙可以进行测试。这样可确保找到尽可能多的错误，然后将其连续交付给生产。

5. DevOps 和持续交付之间有什么区别？

答：DevOps 更像是一种组织和文化方法，可促进工程团队和运营团队之间的协作和沟通。

同时，持续交付是成功将 DevOps 实施到产品开发工作流程中的重要因素。持续交付实践有助于使新发行的版本更加乏味和可靠，并建立更加无缝和短的流程。

DevOps 的主要目的是有效地结合 Dev 和 Ops 角色，消除所有孤岛，并实现独立于持续交付实践的业务目标。

另一方面，如果已经有 DevOps 流程，则连续交付效果最佳。因此，它扩大了协作并简化了组织的统一产品开发周期。

6. 敏捷，精益 IT 和 DevOps 之间有什么区别？

答：敏捷是仅专注于软件开发的方法。敏捷旨在迭代开发，建立持续交付，缩短反馈循环以及在整个软件开发生命周期（SDLC）中改善团队协作。

精益 IT 是一种旨在简化产品开发周期价值流的方法。精益专注于消除不必要的过程，这些过程不会增加价值，并创建流程来优化价值流。

DevOps 专注于开发和部署-产品开发过程的 Dev 和 Ops。其目标是有效整合自动化工具和 IT 专业人员之间的角色，以实现更简化和自动化的流程。