.NET 程序部署注意事项

1. 尽量使用发布后的程序来部署，而不是直接用 Debug 或者 Release 文件夹下的程序，微软对发布程序进行了优化。
2. 在发布的时候点击展开 【文件发布选项】，建议勾选 【启用 ReadyToRun 编译】，它会尝试在发布的时候把部分程序集编译为本地代码，从而提高程序的启动速度。
3. 大部分负载均衡服务器都支持开启把原始的用户 IP 地址放到名字为 X-Forwarded-For 的请求报文头中的设置。在负载均衡服务器上开启这个设置后，我们只要为应用程序安装 ForwardedHeadersMiddleware 中间件，这样 ASP.NET Core 就会把请求报文头中的 X-Forwarded-For （如果有的话）值应用到HttpContext.Connection.RemoteIpAddress 中，因此我们无论是否使用负载均衡服务器，都可以用 HttpContext.Connection.RemoteIpAddress 获取客户端的 IP 地址。当然，使用 HttpContext.Connection.RemoteIpAddress 获取用户 IP 地址的时候，一定要注意 X-Forwarded-For 造假的问题。防范这种问题的手段有两点需要注意：如果网站应用程序直接面对用户请求，而没有使用负载均衡服务器的话，就不要启用 ForwardedHeadersMiddleware 中间件；在面向最终用户的负载均衡服务器上，请设置忽略客户端请求报文头中的 X-Forwarded-For 。
4. 程序更新：.NET 程序在运行时会锁定 DLL 等文件，因此如果我们有新版网站应用程序要替换在运行中的版本的时候，操作系统会提示“文件被占用”，从而无法完成替换。如果网站部署在 IIS 中，有两种解决方法。一种方法是我们编写一个内容包含“网站正在更新”的 HTML 文件，文件名是 app_offline.htm，然后把这个文件放到网站的根目录下，当 IIS 检测到这个文件以后，就会关闭网站，我们就可以覆盖程序进行更新了。在更新期间，对于新的请求，IIS 会把 app_offline.html 的内容返回给客户端，因此访问者看到的就是“网站正在更新”这样的提示信息。当网站完成更新后，我们删除 app_offline.html 即可，下一个请求到来后 IIS 将自动启动并应用。这种方法比较适合企业内部应用等允许有下线时间的系统，对于互联网网站等不允许有下线时间的系统，我们可以使用 .NET 6 新增的“影子拷贝”（shadow-copying），它允许我们在程序运行时替换程序集，具体用法请查看微软文档。无论采用哪种方式更新网站，只要我们用负载均衡服务器把用户请求转发给多台网站服务器，就会存在新旧版网站同时运行的短暂时间，甚至有可能来自同一个用户的属于同一个业务流程的两个连续的请求分别被新旧版两个程序处理。因此我们在编写新版系统的时候，要考虑规避这样的短时间内新旧版程序共存导致的逻辑混乱的问题。
5. 要启用负载均衡服务器。这样恶意攻击者就只知道负载均衡服务器的 IP 地址，而不知道 Web 服务器的 IP 地址，降低恶意攻击者直接访问 Web 服务器的安全风险。
6. 要启用 WAF（Web application firewall，Web 应用程序防火墙），WAF 可以阻挡相当一

部分潜在的网络攻击。

7. 数据库服务器只允许 Web 服务器的 IP 地址访问；数据库服务器一定要设置定时自动备份机制，并且把备份文件异地保存，以便在出现问题时及时恢复数据。
8. 严格区分开发环境和生产环境，增强生产环境的访问权限管理，避免开发人员直接访问生产环境的服务器。
9. 对开发人员的代码进行审查，特别要防范 CSRF（cross-site request forgery，跨站请求伪造）、XSS（cross site scripting，跨站脚本攻击）、SQL 注入漏洞、请求重放攻击等。
10. 防范关键业务数据的“可预测性”。如果我们用自动增长值作为订单的主键的话，竞争对手就可以从订单的主键推测出我们的业务量，而且竞争对手也可以通过对订单主键值的简单递增遍历来批量抓取数据。这个问题的解决方案就是用 Guid 等不可预测的值作为主键值。
11. 避免服务器端发送给客户端的报错信息造成的泄密，尽量不要把服务器内部的细节发送给客户端。比如我们不能直接把服务器端的异常堆栈发送到客户端，因为异常堆栈中可能包含系统重要的技术秘密，甚至可能包含数据库的连接配置等信息。