این مشکل معمولاً به دلیل تفاوت‌هایی بین این دو محیط است. برای مثال، تفاوت در نسخه‌های نرم‌افزار، وابستگی‌ها، تنظیمات سیستم‌عامل، متغیرهای محیطی یا حتی داده‌هایی که برنامه با آن‌ها سروکار دارد. برای دیباگ کردن چنین مشکلی، اولین گام شناسایی این تفاوت‌هاست.

یکی از این کارها مقایسه‌ی دقیق محیط توسعه و محیط اجرا است. مثلاً اگر در محیط توسعه از پایگاه‌داده‌ی محلی استفاده شده ولی در سرور از یک پایگاه‌داده‌ی از راه دور با نسخه‌ی متفاوت، باید بررسی کنیم که تنظیمات اتصال، نسخه‌ها و نوع داده‌ها یکسان هستند یا نه. مثلا فرض کنید در برنامه‌مان از کتابخانه‌ی Python به نام pandas استفاده کرده‌ایم. در محیط توسعه نسخه‌ی 2.1.0 نصب شده، ولی در سرور نسخه‌ی 1.3.0 وجود دارد. حالا اگر از توابع جدیدی مثل convert_dtypes() استفاده کرده باشیم که در نسخه‌ی پایین‌تر وجود ندارد، برنامه در سرور با خطا مواجه می‌شود. راه‌حل این است که با استفاده از فایل‌هایی مانند requirements.txt یا ابزارهایی مثل Docker، مطمئن شویم که نسخه‌ی دقیق همه‌چیز بین دو محیط همسان باشد.

در مرحله‌ی بعد، استفاده از لاگ‌گیری (Logging) بسیار مفید است. لاگ‌ها به ما نشان می‌دهند برنامه دقیقاً در کجا و به چه دلیلی متوقف شده یا خطا داده است. به جای rely کردن صرفاً بر پیام خطا، باید از لاگ‌های دقیق‌تر و حتی ابزارهایی مانند Sentry برای رهگیری خطا در سرور استفاده کنیم.

در نهایت، یکی از بهترین روش‌ها برای جلوگیری از این نوع مشکلات، استفاده از کانتینرها مانند Docker است. با Docker می‌توانیم محیط توسعه را دقیقاً همان‌طور که در سرور قرار است باشد، شبیه‌سازی کنیم تا اختلاف محیطی از بین برود.