# Why Jenkins Should Run as a Non-Root User

## 1. Security

Minimizing privileges: Running Jenkins as root gives it full control over the system. If Jenkins or one of its plugins is compromised, an attacker could gain root access. Running as a non-root user limits this risk.  
  
Safer plugin execution: Jenkins plugins, many of which are developed by third parties, cannot access sensitive system areas when Jenkins runs as a regular user.

## 2. Least Privilege Principle

Jenkins only needs access to specific directories and files. Following the principle of least privilege limits its access, reducing the risk of system-wide impact if compromised.

## 3. Avoid Accidental Damage

Commands or scripts executed by Jenkins as root could unintentionally delete or corrupt critical system files. Running as a non-root user limits such risks to the Jenkins environment only.

## 4. Better Logging and Auditing

Actions performed by the Jenkins user are easily tracked in system logs. This makes it simpler to audit Jenkins activity separately from root-level operations.

## 5. Improved Compatibility

Certain tools used within Jenkins jobs, like Docker or Git, expect to run under a normal user account. Running Jenkins as non-root improves compatibility and integration with such tools.

## Recommended Practice

• Create a dedicated system user for Jenkins (e.g., 'jenkins')  
• Set ownership of Jenkins directories (e.g., /var/lib/jenkins) to this user  
• Use systemd or init scripts to ensure Jenkins runs under the 'jenkins' user