# Migrating to Chainguard Images

Chainguard Container Images are designed to be minimal and to include special features for increased security and provenance attestation. Depending on your current base image and customizations, you may need to make some adjustments when migrating your current workloads to use Chainguard Container Images.

## Important to Know

- Most Chainguard Container Images don't have a package manager or a shell by default. These are distroless images intended to be used as slim runtimes for production environments.
- For every version of an image, a complimentary **standard** image is provided with a shell and the apk package manager. These are identified by the `-dev` suffix and can be customized.
- When possible, we recommend using multistage builds that combine a build stage based on a `-dev` variant and a runtime stage based on a distroless image.

- Chainguard Container Images typically don't run as root, so a `USER root` statement may be required before installing software.
- Chainguard Container Images are based on **apk**. If you're coming from Debian or Ubuntu you'll need to replace `apt` commands with their `apk` equivalents. This also applies for other distros that are not based on **apk**.
- Some images may behave differently than their equivalent in other distros, due to differences in entrypoint and shell availability. Always check the image documentation for usage details.

## Migration Checklist

- ☐ Check the image's overview page on the Images Directory for usage details and any compatibility remarks.
- ☐ Replace your current base image with a standard `-dev` (such as `latest-dev`) variant as a starting point.
- ☐ Add a `USER root` statement before package installations or other commands that must run as an administrative user.
- ☐ Replace any instances of `apt install` (or equivalent) with `apk add`.
- ☐ Use `apk search` on a running container or the APK Explorer tool to identify packages you need – some commands might be available with different names or bundled with different packages.
- ☐ When copying application files to the image, make sure proper permissions are set.
- ☐ Switch back to a nonroot user so that the image does not run as root by default.
- ☐ Build and test your image to validate your setup.
- ☐ Optional: migrate your setup to a multi-stage build that uses a distroless image variant as runtime. Our Getting Started with Distroless guide has detailed information on how to work with distroless images and multi-stage builds.

For detailed migration guidance, please refer to our Migration Docs on Chainguard Academy. For troubleshooting, check our Debugging Distroless Images resource.