Find which Linux process is using a port

An address-already-in-use error means another socket conflicts with the requested bind. It does not tell you which service should be stopped. These checks apply to Debian or Ubuntu administrators investigating an application startup failure. They inspect listeners and deployment definitions without terminating processes or modifying firewall rules.

Capture the exact bind failure

Record the address, port, protocol and timestamp from the application's log. A TCP listener and a UDP listener on the same number are different sockets. A loopback bind and an all-interface bind may also interact differently. Check whether the application is starting twice through a panel, shell session or duplicate service unit.

```
sudo ss -ltnp
sudo ss -lunp
```

The first output lists listening TCP sockets; the second lists UDP sockets. Administrative access usually gives a more complete process view. Find the relevant port rather than assuming the application's name will appear. A process may use a generic interpreter or wrapper name.

Identify the owner before acting

Note the PID and inspect it with `ps -p PID -o pid,ppid,user,etime,comm`, replacing PID with the recorded number. Avoid sharing command lines that contain credentials. Use `systemctl status PID` when supported to find the unit associated with that process, and check its logs.

A reverse proxy may intentionally own ports 80 and 443 while the backend uses a local high port. Starting a second web server on the public port is then the configuration error. Likewise, a management panel may own the process lifecycle. Killing it manually can trigger an immediate restart and hide the deployment mistake.

Check containers and namespaces

If Docker is installed and the service is containerized, inspect `docker ps` and the affected container's published ports. A host-side proxy process may represent a container mapping. In host-network mode the application shares the host network namespace, while an ordinary bridged container has a separate internal view.

Do not stop every container to discover the owner. Compare the deployment definition with the host listener. A mapping can conflict even when the container's internal port looks different. Preserve volume and dependency information before any planned recreation.

Choose one intentional owner

Decide whether the existing service is legitimate, obsolete or misconfigured. If both applications are required, assign a documented unused port or route through the existing proxy. Check the new port against allocations and dependent clients before changing it. An apparently free number is not evidence that the hosting plan has allocated it.

Save the previous configuration and change only the affected application. If a stop is necessary, use its service manager or panel in a maintenance window. Do not send a kill signal to an unknown PID, especially a database or storage daemon.

Verify the complete path

Start the intended service, repeat the listener checks and make a normal application request from an authorized client. Review logs for both startup and successful traffic. If the change fails, restore the saved configuration and original service arrangement. Keep a note of the final address, protocol and owning unit so the same conflict is easier to diagnose next time.

Related instructions

Investigate systemd services. Diagnose network timeouts.

Reference: Official documentation. Check the documentation installed on your own operating-system release when an option differs.

Revision #1
Created 2026-11-09 10:42:43 UTC by CloudExa Hosting Admin
Updated 2026-11-09 10:42:43 UTC by CloudExa Hosting Admin