Find, Evaluate, and Deploy Self-Hosted Services

Linux Training Academy

Lesson Overview

- Identify alternatives to costly or third-party services
- Evaluate options based on your specific needs
- Deploy selected alternative services

Why Choose Self-Hosted Alternatives

- Self-hosting gives you **control** over your services
- **Avoid monthly fees** for tools you could run yourself
- Protect against pricing changes or service shutdowns

A Quick Note Before We Dive In

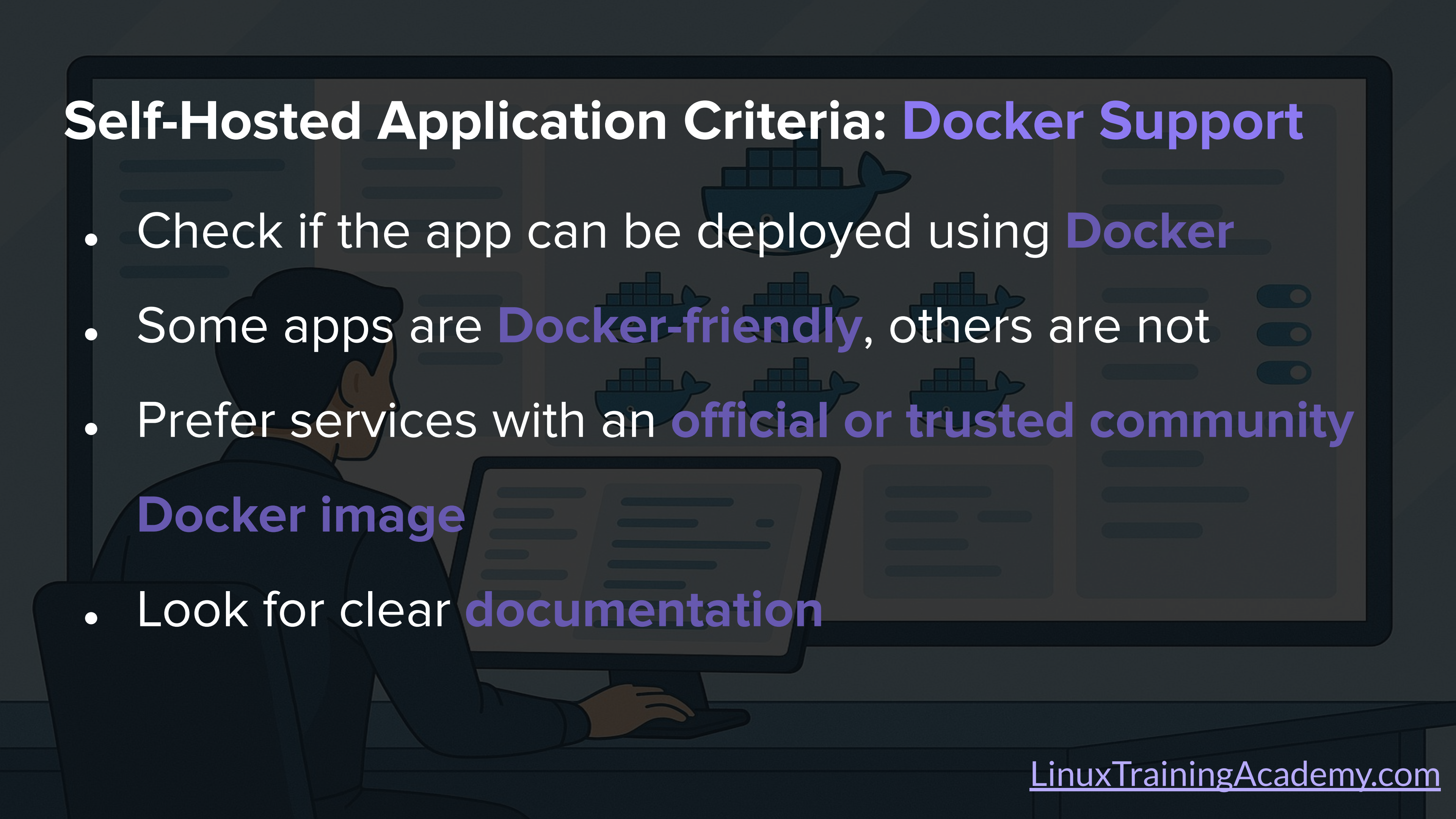
- The goal is to help you **build confidence**
- Learn to explore, evaluate, and deploy services independently
- Focus on **your unique needs**
- Gain skills to assess tools without future tutorials

Another Quick Note Before We Dive In

- **Links will be included** for everything
  - Use them to explore resources on your own
- The Internet is constantly changing
  - Resources may look different, change, or disappear
- Focus on learning and applying the **concepts**

Self-Hosted Application Criteria: Features

- Functionality is the **top priority**
- Does it meet your specific **requirements**?
- Rule out apps that **lack core features** you rely on
- Prioritize software that does exactly what **you need**



Self-Hosted Application Criteria: **Docker Support**

- Check if the app can be deployed using **Docker**
- Some apps are **Docker-friendly**, others are not
- Prefer services with an **official or trusted community Docker image**
- Look for clear **documentation**

Self-Hosted Application Criteria: Activity

- Check if the project is actively maintained
- Look for recent updates and release activity
- Most open-source projects are on GitHub
- Search: software name + "GitHub" or "source code"
- Frequent commits = active development

Self-Hosted Application Criteria: Documentation

- Documentation may be in the source code or on a separate site
- No documentation is a **red flag**
- Good docs help you **get started fast** and solve issues quickly
  - Installation
  - Configuration
  - Troubleshooting

Deploying an Application Using Docker

- Find or create a **compose.yaml** file
- Adjust host ports and deploy
- To connect with Tailscale, add **TSDProxy labels**

Locating Docker Compose Files

- Docs may refer back to the repository for setup files
- Look for **compose.yaml** or **docker-compose.yml**
- Sometimes Docker instructions are missing from the docs
- Check the repo – the YAML file may be there anyway

Using Docker Hub for Official Images

- A **Dockerfile** usually means an official image exists
- Visit **hub.docker.com** and search the software name
- Docker Hub often includes:
  - Usage instructions
  - docker run examples
  - Sample compose files

Converting Docker Run Commands

- No compose file? Look for a docker run command
- You can **convert** docker run to compose.yaml
- Use utilities like **IT-Tools** (covered in a previous lesson)
- Ask AI tools to **generate** a compose file for you

Search Engine and Community Resources

If you’re stuck, try a search engine:

“[app name] docker compose”

“[app name] compose.yaml”

“docker-compose.yml for [app name]”

Check the **Portainer Community Templates Site**:

https://portainer-templates.as93.net

Customizing Compose Files for Your Environment

- Search for example compose.yaml files for your app
- **Don't assume they work as-is** – modify to fit your setup
- Update **host ports** to avoid conflicts
- Adjust bind mount **paths** to reflect your environment
- Use online files as **starting points**, not final solutions

Understanding Docker Image Documentation

- Always read the Docker image **documentation**
- Docs can be on Docker Hub, GitHub, or the project’s website
- Look for **environment variables** – required or optional
- Check **volume mappings** for persistent storage

  /config in container ↔ /opt/docker/app-name/config on host
- Find the internal container port to map to a host port

Moving Forward with Confidence

- You now have the **confidence** to deploy any service
- You’ve gained skills and knowledge to meet your goals
- You've already handled the hardest parts of self-hosting
- Your foundation is strong and secure
- You’re ahead of most people trying to self-host today

Lesson Recap

- How to **discover** self-hosted alternatives to current services
- Awesome Self-Hosted and selfh.st **directories**
- Use of search engines, AI tools, and community forums
- How to evaluate applications before deploying:
  - Features, Docker support, maintenance, community, documentation
- How to find and customize a **compose.yaml** file
- You can now deploy services tailored to **your** environment