Lab: Container-Supported Development

- Launch the lab
- What you'll learn
- Modules

Use containers to run the services your app depends on — databases, caches, message queues — without installing anything locally. This lab walks through running PostgreSQL in a container, writing a `compose.yaml` your whole team can share, and adding a pgAdmin visualizer to the dev stack.

Launch the lab

{{< labspace-launch image="dockersamples/labspace-container-supported-development" >}}

What you'll learn

By the end of this Labspace, you will have completed the following:

- Run a PostgreSQL database in a container with no local installation
- Use bind mounts to seed a database with schema and initial data at startup
- Write a `compose.yaml` that codifies the entire dev stack for the team
- Add a pgAdmin container to visualize and inspect the database
- Understand how containerized dev stacks reduce onboarding time and environment drift

Modules

#	Module	Description

1	Introduction	Meet the sample app and understand the container-supported development approach
2	Running a Containerized Database	Start PostgreSQL, connect the app, and seed the database using bind mounts
3	Making Life Easier with Compose	Replace `docker run` commands with a shared `compose.yaml`
4	Adding Dev Tools	Add pgAdmin to the Compose stack for database visualization
5	Recap	Review key takeaways and explore related guides

□□□ #2
□ □□□ □□□ 2026-10-31 18:46:30 UTC
□ □□□ □□□ 2026-10-31 18:46:31 UTC