

# Scaling

- If your application is **stateless** you can horizontally scale it
  - Stateless = your app doesn't have a **state**, it doesn't **write** any **local file** / keeps local session.
  - All traditional database ( MySQL, Postgres) are **stateful**, they have database files that can't be split over multiple sessions.
- Most **Web Application** can be made stateless:
  - **Session Management** needs to be done outside the container ( MemCache, Redis etc)
  - Any files that need to be saved **can't be saved locally** on the container

# Scaling

- Our example app is **stateless**, if the same app would run multiple times, it doesn't change state.
- Later in this course I'll explain how to use **volumes** to still run stateful apps
  - Those stateful apps can't horizontally scale, but you can run them in single container & vertically scale ( allocate more CPU/ Memory/ Disk )

# Scaling

- Scaling in Kubernetes can be done using the **Replication Controller**
- The Replication Controller will **ensure** a specified number of **pod replicas** will run at all times
- A Pods created with Replication Controller will **automatically** be **replaced** if they fail, get deleted, or are terminated.
- Using the Replication Controller is also **recommended** if you just want to make sure **1 pod** is always running, even after reboot.
  - You can then run a replication controller with just **1 replica**
  - This makes sure that the pod is always running.

# Demo Placeholder

- Scaling