



#### KN05: Arbeit mit Speicher

- A) Bind mounts (40%)
- B) Volumes (30%)
- C) Speicher mit docker compose (30%)

## KN05: Arbeit mit Speicher

---

Beachten Sie die [allgemeinen Informationen zu den Abgaben](#).

### Grundlagen für diesen Auftrag:

- [TBZ: Arbeit mit Volumes](#)
- [Docker: Storage](#)
- [Praxistipps.de: Bildschirmaufnahme in Windows 10](#)

Sie werden im Folgenden die drei Speicherarten testen.

### A) Bind mounts (40%)

---

Sie werden hier zeigen, dass Ihr Speicher von ihrem Host mit dem Container geteilt wird. In der Theorie haben Sie gelernt für welche Zwecke dieser Speicher nützlich sein kann. Sie simulieren den Fall, dass Sie eine Applikation programmieren und regelmässig Änderungen im Code vornehmen und auf dem Container testen müssen.

Erstellen Sie ein Container (z.B. mit dem nginx Basis-Image) und fügen Sie ihm ein **Bind Mount** hinzu. Erstellen Sie jetzt ein Bash-Skript auf Ihrem Host-System (also wahrscheinlich in Ihrem Windows), welches eine einfache Ausgabe von Inhalt macht. Stellen Sie sicher, dass Ihr Code einzigartig ist und keine Kopie von Ihren Kollegen. In Ihrem Container rufen Sie das Skript auf, so dass die Ausgabe angezeigt wird. Ändern Sie das Skript nun in Ihrem Host-System und führen Sie es wieder auf dem Container aus. Die Änderungen sollten ebenfalls sichtbar sein.

#### Abgabe:

- Liste der Befehle die notwendig waren, um Container mit Volumen zu starten.
- Erstellen Sie einen Screencast, der den beschriebenen Prozess zeigt. Testen Sie ihn aber erst, bevor Sie den Screencast erstellen (Link in Grundlagen-Teil).

### B) Volumes (30%)

---

Sie werden hier zeigen, dass Ihre Container die gleichen Volumen verwenden können. Dies kann in einer Applikation notwendig sein.

Erstellen Sie **zwei** Container (z.B. mit dem nginx Basis-Image) und fügen Sie beiden das **gleiche Named Volume** hinzu. Öffnen Sie die Konsolen beider Instanzen und schreiben von beiden Container in die gleiche Datei und lesen Sie die Inhalte auf der anderen Seite jeweils wieder aus.

Sie können `cat` zum Lesen und `echo` mit Umleitung der Ausgabe in eine Datei (`>>`) zum Schreiben benutzen. Sie hatten die entsprechenden Befehle im Modul m122.

**Abgabe:**

- Liste der Befehle die notwendig waren, um Container mit Volumen zu starten.
- Erstellen Sie einen Screenshot, der den beschriebenen Prozess zeigt. Testen Sie ihn aber erst, bevor Sie den Screenshot erstellen (Link in Grundlagen-Teil).

## C) Speicher mit docker compose (30%)

---

Erstellen Sie ein Container mit docker compose und fügen Sie dem Container die **drei** verschiedenen Speichertypen (bind mount, named volume und tmpfs) hinzu. Gehen Sie wie folgt vor:

- Erstellen sie ein *Named Volume*, indem Sie [das entsprechende Top Level Element](#) verwenden.
- Erstellen Sie **zwei** ein nginx-container.
- Fügen Sie das erstellte named volume dem **ersten** Container hinzu unter [Verwendung der Long Syntax](#).
- Fügen Sie das erstellte named volume dem **zweiten** Container hinzu unter [Verwendung der Short Syntax](#).
- Fügen Sie dem **ersten** Container ein bind mount hinzu.
- Fügen Sie dem **ersten** Container ein tmpfs mount hinzu.

**Abgaben:**

- Auszug mit dem Befehl `mount` im ersten Container, der zeigt, dass alle drei Speichertypen hinzugefügt wurden.
- Auszug mit dem Befehl `mount` im zweiten Container, der zeigt, dass der Speichertyp hinzugefügt wurde.
- docker compose Datei (yaml).