



[TOC]

## KN03: Netzwerk, Sicherheit

---

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

### Grundlagen für diesen Auftrag:

- [TBZ: Netzwerk](#)
- [TBZ: Sicherheit](#)
- [CloudKul: Understanding Communication Between Docker Containers](#)

### A) Eigenes Netzwerk (100%)

In Auftrag KN02 verwendeten Sie einen Link, damit die Container miteinander kommunizieren können. Verwendet wird dabei die *default bridge* und Ihre Kontrolle über die Kommunikationskanäle ist begrenzt. Sie haben ja die Möglichkeit eigene Netzwerke zu erstellen (die entsprechenden Befehle finden Sie in den Grundlagen). Folgend werden Sie die Abgrenzung von Netzwerken testen:

Das Zielsystem besteht aus vier Containern, die auf dem Image busybox ([BusyBox: The Swiss Army Knife of Embedded Linux](#)) basieren. Zwei der Container sollen im default bridge Netzwerk angeschlossen sein. Die anderen zwei Container schliessen Sie in einem user-defined Netzwerk mit Namen **tbz** an.

Wenn Sie nichts anderes angeben, verwaltet Docker für Sie die IP-Ranges und IP-Adressen der Container. Folgendes Bild zeigt Ihnen das Zielsystem - inkl. IP-Adressen. Weil die IP-Adressen mit drin sind im Bild zur Verdeutlichung, kann Ihr effektiv aufgebautes System bei den IP-Adressen Abweichungen aufzeigen. Das ist in Ordnung.

```
skinparam monochrome true
skinparam packageStyle rectangle
hide circle
hide empty members

class "host" {
    "172.17.0.1, 172.18.0.1"
}

package "default bridge (172-*-*-/16)" {
    class "busybox1" {
        "172.*.*.*"
    }
    class "busybox2" {
```

```
    "172.*.*.*
  }

  "busybox1" -- "busybox2" : ""
  "busybox1" -- "host" : ""
  "busybox2" -- "host" : ""
}
package "m347-net ('172-18-0-0/16')" {
  class "busybox3" {
    "172.18.0.2
  }
  class "busybox4" {
    "172.18.0.3
  }

  "busybox3" -- "busybox4" : ""
  "busybox3" -- "host" : ""
  "busybox4" -- "host" : ""
}
```

Bauen Sie nun selbst das Zielsystem bei sich auf dem Rechner auf. Sie sollen auch den gleiche **IP Range 172.18.0.0/16 verwenden**. Verwenden Sie das **busybox-Image**. Arbeiten Sie mit dem **docker network create**-Befehl, um das tbz-Netzwerk anzulegen. Das *default bridge* Netzwerk können Sie nicht ändern. Erstellen/Starten Sie die busybox-Container.

Die Netzwerk-Tools sind im **busbox bereits installiert**.

Um die Netzwerkkonfiguration bei Linux zu analysieren und die Verbindungen zu testen benötigen Sie folgende Befehle.

Befehl	Beispiel	Bedeutung
ifconfig	ifconfig -a	Der Befehl zeigt die Konfiguration der Netzwerk-Interfaces. Mit dem Parameter -a zeigen Sie alle Netzwerkinterfaces an. Für diese Aufgabe reicht es aber wenn Sie nur ifconfig eingeben.
ping	ping busybox3, ping 172.17.0.1	Mit dem ping-Befehl können Sie andere Geräte (in unserem Fall Container) innerhalb des Netzwerks anpingen. Wenn die Container erreichbar sind, werden diese auf die Ping-Anfragen antworten und das ping-Programm zeigt Ihnen an, in welcher Zeit geantwortet wurde.

Verwenden Sie die Befehle um die Unterschiede und Gemeinsamkeiten in den Netzwerken zu finden:

- 1. Welche IP-Adressen haben busybox1, busybox2, busybox3 und busybox4 erhalten? Diese Aufgabe können Sie auch mit **docker inspect** lösen
- 2. Starten Sie eine interaktive Session auf **busybox1** und geben Sie folgende Befehle ein, resp. finden Sie die korrekten Befehle:
  - 1. Welcher Default-Gateway ist eingetragen? Welcher Container hat den gleichen?
  - 2. ping busybox2
  - 3. ping busybox3
  - 4. ping IP-von-busybox2

5. ping IP-von-busybox3
3. Starten Sie eine interaktive Session auf **busybox3** und geben Sie folgende Befehle ein:
  1. Welcher Default-Gateway ist eingetragen? Welcher Container hat den gleichen?
  2. ping busybox1
  3. ping busybox4
  4. ping IP-von-busybox1
  5. ping IP-von-busybox4

**Abgaben:**

- Screenshots der Befehle und deren Resultate. Achtung: Sie können Befehle zusammenfassen. Führen Sie die Befehle in einer sinnvollen Reihenfolge aus.
- Erklären Sie die Gemeinsamkeiten und Unterschiede. Wie kommen die Zustände und was ist Ihre Schlussfolgerung.
- Betrachten Sie nun KN02.
  - In welchem Netzwerk befanden sich die beiden Container?
  - Wieso konnten die miteinander reden?