

Private IPv4-Adressen im Campusnetz der TU Dresden

12.03.2026 20:16:57

FAQ-Artikel-Ausdruck

Kategorie:	Datennetz::Zentrale IP-Adressverwaltung	Bewertungen:	2
Status:	öffentlich (Alle)	Ergebnis:	100.00 %
Sprache:	de	Letzte Aktualisierung:	13:05:01 - 11.02.2026

Schlüsselwörter

NAT Network Address Translation IPv4 IPv6 Docker VirtualBox HyperV

Lösung (öffentlich)

Aufgrund der Knappheit öffentlicher IPv4-Adressen, erfolgt der Betrieb von Infrastrukturen zunehmend mit privaten IPv4-Adressen, z.B. in Kombination mit Network Address Translation (NAT). Auch im Netzwerk der TU Dresden werden private IP-Adressen verwendet und geroutet.

Um IP-Adress-Konflikte zu vermeiden ist es daher wichtig vorsichtig bei der Verwendung zu sein.

Falls Sie auf einzelnen Systemen ebenfalls private IP-Adressen verwenden. z.B. für Container (Docker, Podman, Kubernetes, LXD) oder virtuelle Maschinen (VirtualBox, VMware Workstation Player, Parallels, HyperV), nutzen Sie bitte den explizit für diesen Zweck vorgesehenen Netzbereich 172.17.0.0/16.

Anderenfalls kann es sein, dass Ihr Dienst für Nutzer:innen aus TU-internen Netzen nicht nutzbar ist oder Sie TU-interne Dienste nicht verwenden können. In diesem Artikel wird eine Beispielkonfiguration für Docker bereitgestellt.

Sollten Sie größere Infrastrukturen mit privaten IP-Adressen betreiben wollen, für die das genannte Subnetz nicht ausreicht, kontaktieren Sie bitte den ServiceDesk. Mögliche Probleme durch Verwendung von bereits im Campus genutzten privaten IPv4-Adressen Beispiel 1: Ein Nutzer versucht mit der Adresse 172.31.61.61/24 aus dem TU-intern verwendeten privaten IPv4-Subnetz 172.31.61.0/24 einen Dienst zu erreichen. Auf der VM des Dienstes verwendet Docker allerdings Subnetz 172.31.0.0/16, dass die IP-Adresse des Nutzers enthält. Die Anfragen des Nutzers erreichen den Dienst und werden dort beantwortet. Allerdings versucht das Betriebssystem der VM die Antworten in das von Docker verwendete Subnetz zu routen, da diese Route spezifischer ist als die Standardroute in den Campus wo der Nutzer zu erreichen ist. Der Nutzer erhält keine Antworten auf seine Anfragen.

Beispiel 2: Ein Nutzer verwendet auf seinem Laptop Docker zur Softwareentwicklung. Dabei ist das Subnetz 172.31.0.0/16 im Einsatz. Der Nutzer versucht nun einen TU-internen Dienst zu verwenden, dessen Adresse im Subnetz 172.31.61.0/24 liegt. Das Betriebssystem des Laptops versucht die Anfrage des Nutzers in das von Docker verwendete Subnetz zu routen, da die bekannt Route dorthin spezifischer ist als die Standardroute in den Campus. Der Nutzer erhält keine Antworten auf seine Anfragen. Beispielkonfiguration für Docker

File: /etc/docker/daemon.json

```
{
  "bip": "172.17.0.1/24",
  "default-address-pools": [
    {
      "base": "172.17.0.0/16",
      "size": 24
    }
  ]
}
```