

Vorbereitung der Seminar-Umgebung (Docker und Kubernetes)

Hard- und Softwarevoraussetzungen für Seminare zu den Themen Docker und Kubernetes (Mindestanforderungen)

Basisinstallation

Unsere Anleitung *Lab Requirements für Java-Seminare* beschreibt die Hard- und Softwarebasis für unsere Seminare. Für Docker und Kubernetes sollten abweichend davon mindestens 16 GB Hauptspeicher (besser 32 GB) und 50 GB (besser 100 GB) freier Festplattenspeicher verfügbar sein.

Bei Einsatz von Windows benötigen Sie MS Windows 10 oder höher. Die Nutzung von WSL2 (*Windows Subsystem for Linux 2*) ist empfehlenswert und für die Home-Version sogar zwingend notwendig.

Software

Zusätzlich zu den in der Basisinstallation genannten Werkzeugen benötigen wir hier:

- Visual Studio Code (<https://code.visualstudio.com/download>)

Zur Bearbeitung von Dockerfiles und Kubernetes-Manifesten werden die Extensions "Docker" und "Kubernetes" benötigt.

- Bash

Grundsätzlich ist auch die Nutzung der Windows-Eingabeaufforderung möglich, allerdings steht mit der Bash eine deutlich leistungsfähigere Kommando-Umgebung zur Verfügung. Viele Beispielskripte der Seminarprojekte sind in Bash-Syntax verfasst.

Bewährt hat sich die Bash-Implementierung, die mit Git for Windows ausgeliefert wird (<https://git-scm.com/downloads>). Wählen Sie bei der Installation bitte aus, dass

- die Unix/Linux-Tools in den Suchpfad aufgenommen werden,
- und das Windows-Terminal statt MinTTY genutzt wird.

Wenn Sie *Chocolatey* zur Installation nutzen, können Sie diesen Befehl verwenden:

```
choco install -y git --params "/GitAndUnixToolsOnPath /WindowsTerminal"
```

- Docker und Kubernetes

Je nach Zielumgebung können Sie wie folgt vorgehen:

- MS Windows

Gute Erfahrungen haben wir mit *Docker Desktop* gemacht. Docker Desktop ist eine Docker-Implementierung für Windows (auch für Mac erhältlich), in die Kubernetes bereits integriert ist. Befolgen Sie die Installationsanweisungen auf <https://docs.docker.com/desktop/windows/install/>. Empfehlenswert ist die Nutzung des *Windows Subsystem 2 for Linux*, das für Windows 10 ab der Version 2004 verfügbar ist (Home und Pro). Sollten Sie WSL 2 noch nicht zur Verfügung haben, benötigen Sie ein Windows 10 Pro mit aktiviertem Hyper-V-Feature.

Installieren Sie bitte eine aktuelle Version. Derzeit führen wir die Seminare mit der Version 4.5.1 oder höher durch. Kontaktieren Sie uns bitte, wenn Sie eine ältere Version einsetzen möchten.

Docker Desktop ist in den aktuellen Versionen nicht mehr allgemein kostenfrei. Für Ausbildungszwecke (oder für Unternehmen mit weniger als 250 Mitarbeitenden) ist die Nutzung ohne Gebühr gestattet.

Sollten Sie die Installation von Software für Ihre Windows-Rechner mit Hilfe von Chocolatey (<https://chocolatey.org/>) vornehmen, können Sie die oben aufgeführten Tools mit dem folgenden Befehl installieren:

```
choco install -y docker-desktop
```



Prüfen Sie nach der Installation auf jeden Fall die ordnungsgemäße Funktion der Umgebung:

- Lassen Sie in der Bash/Eingabeaufforderung den Befehl `docker version` ablaufen. Er muss Versionsangaben für Client und Server ausgeben – ohne Fehlermeldungen.
- Aktivieren Sie Kubernetes in den Settings von Docker Desktop (zu erreichen über einen Rechtsklick auf das Wal-ähnliche Symbol in der Statuszeile bzw. im System Tray. Lassen Sie dann in der Bash/Eingabeaufforderung den Befehl `kubectl version` ablaufen. Er muss analog zu oben Versionsangaben für Client und Server ausgeben – wiederum ohne Fehlermeldungen.

◦ macOS

Auch hier empfehlen wir *Docker Desktop* (vgl. die obigen Ausführungen für Windows).

Sie können zur Softwareinstallation *Homebrew* (<https://docs.brew.sh/>) verwenden. Dessen Installation wird in <https://docs.brew.sh/Installation> beschrieben.

Danach ist die Installation von *Docker Desktop* so möglich:

```
brew install --cask docker
```



Prüfen Sie nach der Installation auf jeden Fall die ordnungsgemäße

Funktion der Umgebung:

- Lassen Sie in der Bash/Eingabeaufforderung den Befehl `docker version` ablaufen. Er muss Versionsangaben für Client und Server ausgeben – ohne Fehlermeldungen.
- Aktivieren Sie Kubernetes in den Settings von Docker Desktop (zu erreichen über einen Klick auf das Wal-ähnliche Symbol in der Menüleiste). Lassen Sie dann in der Bash/Eingabeaufforderung den Befehl `kubectl version` ablaufen. Er muss analog zu oben Versionsangaben für Client und Server ausgeben – wiederum ohne Fehlermeldungen.

- Linux (Debian, Ubuntu o. ä.)

Installieren Sie *Docker* aus den offiziellen Paketquellen:

```
apt-get update
apt-get install -y docker.io
apt-get install -y docker-compose
```

Fügen Sie die User, die *Docker* nutzen sollen, der Gruppe `docker` hinzu:

```
adduser someUserId docker
```

Eventuell müssen Sie sich neu anmelden oder sogar den Rechner neu starten, damit diese Gruppenzuordnung wirksam wird.

Zur Installation von Kubernetes *k3sup* (<https://github.com/alexellis/k3sup>). Damit ist eine einfache Installation der Kubernetes-Distribution *K3s* möglich. Führen Sie die folgenden Befehle als normaler User (nicht `root` aus):

```
curl -sL https://get.k3s.io | INSTALL_K3S_EXEC="--disable traefik" sh -
mkdir ~/.kube
chmod 700 ~/.kube
k3sup install --local --local-path .kube/config --k3s-extra-args='--docker'

export KUBECONFIG=~/.kube/config
```

Den letzten Befehl (`export ...`) tragen Sie am besten in die Datei `~/.bashrc` ein, damit er bei einem erneuten Start der Shell wieder ausgeführt wird.



Prüfen Sie nach der Installation auf jeden Fall die ordnungsgemäße Funktion der Umgebung:

- Lassen Sie in der Bash/Eingabeaufforderung den Befehl `docker version` ablaufen. Er muss Versionsangaben für Client und Server

ausgeben – ohne Fehlermeldungen.

- Lassen Sie dann in der Bash/Eingabeaufforderung den Befehl `kubect1 version` ablaufen. Er muss analog zu oben Versionsangaben für Client und Server ausgeben – wiederum ohne Fehlermeldungen.