A close-up of water droplets

Description automatically generated with low confidence**Betriebsysteme Projekt WS22/23**

**Thema 2:**

**Clustering mit Docker**

**Dokumentation**

**Von Ahmed Abdaal und Jenny Rötzer**

**Inhaltsverzeichnis**

***1. VM einrichten***

***2. Installation von Docker***

***3. NFS***

***4. Portainer***

***5. Docker Swarm einrichten***

***6. ???***

***7. ???***

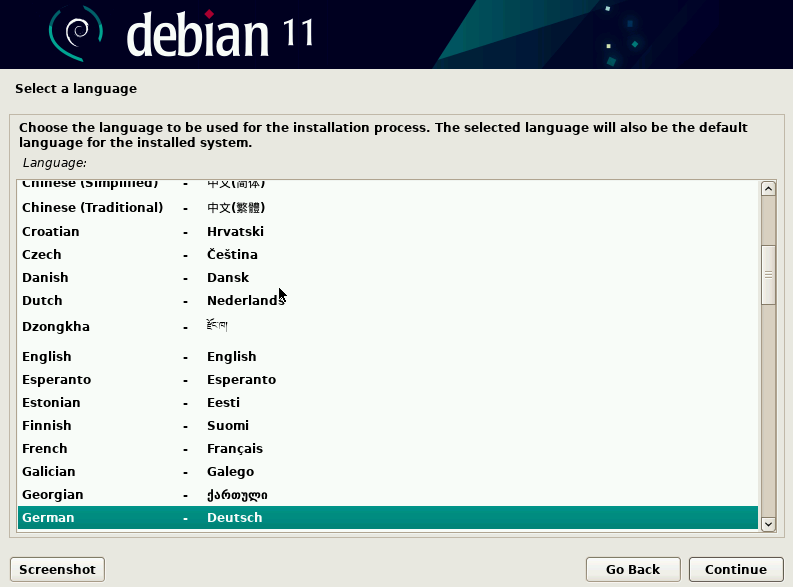
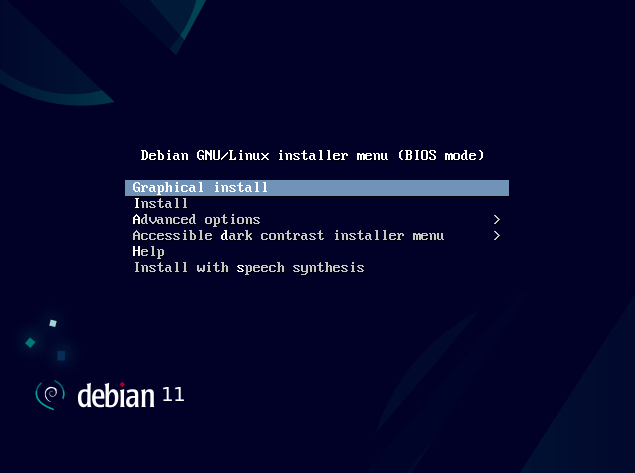
***8. Test***

***9. Quellen***

***Alle lila-gefärbten Themen kannst du dann ausformulieren und erklären.***

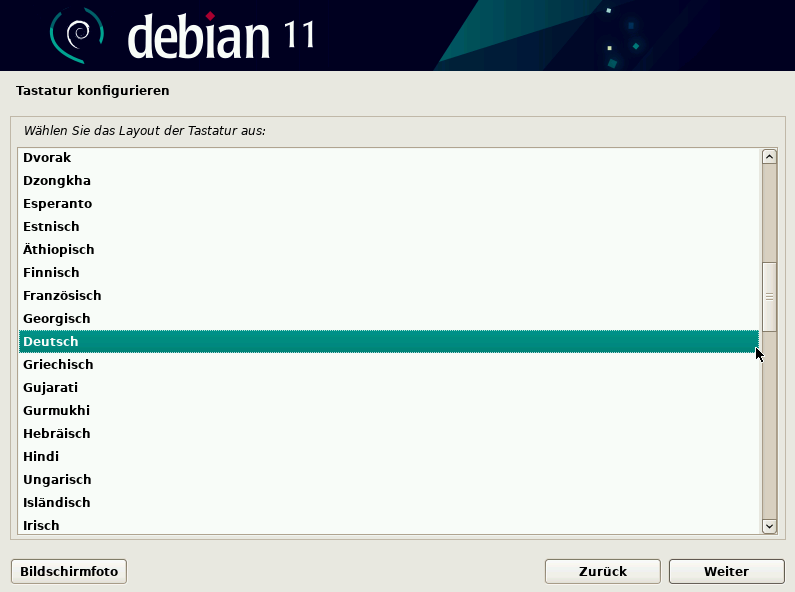
***Du kannst entscheiden ob du Portainer benutzen willst oder Visualizer. Du kannst auch selber entscheiden auf welchen Platz der Stichpunkt kommt, also ob es vor NFS kommt oder danach.***

***Bei den Fragezeichen schauen wir uns erstmal die Lösung von den Dude an, damit wir was checken und dann schreiben wir was rein.***

**1. VM einrichten**

**1. Schritt: „Graphical Install“ Option auswählen und mit Enter bestätigen**

**2. Schritt: Die bevorzugte Sprache auswählen und auf Continue drücken**

**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

**4. Schritt: Die Tastaturbelegung auswählen**

**3. Schritt: Das Land auswählen und bestätigen**

**Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung**

**6. Schritt: Ein Domain-Name vergeben**

**5. Schritt: Einrichtung des Netzwerks und ein Rechnername vergeben**

**Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung**

**8. Schritt: Name des ersten Benutzers eingeben**

**7. Schritt: Ein sicheres Kennwort für den Benutzer eingeben**

**Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung**

**9. Schritt: Benutzername vergeben, meistens wird der Vorname vom Benutzer gewählt**

**10. Schritt: Ein sicheres Passwort für den Benutzer eingeben**

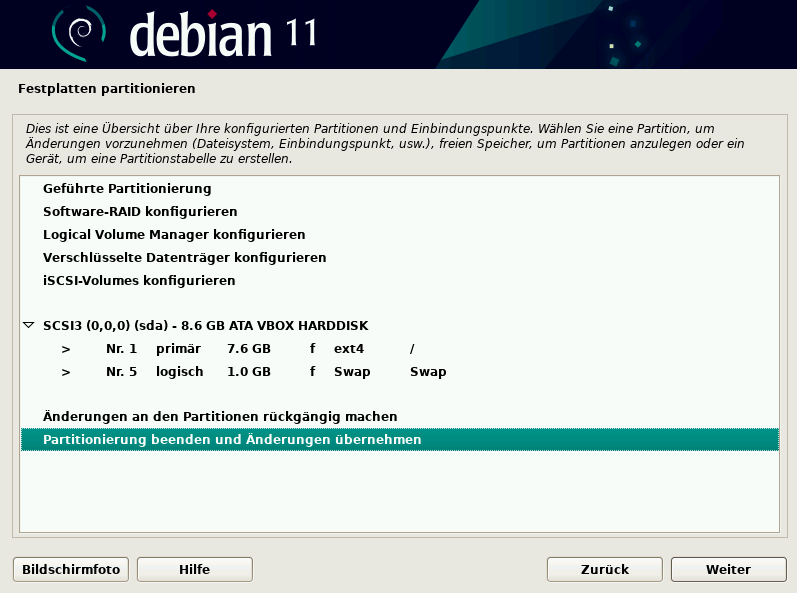
**Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung**

**11. Schritt: Die vollständige Festplatte für die Partitionierung auswählen**

**12. Schritt: Die aufgeführte Auswahl mit Weiter bestätigen**

**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

**14. Schritt: 2. Option „Partitionierung beenden und Änderungen übernehmen“ auswählen und auf Weiter drücken**

**13. Schritt: 1. Option mit „Alle Dateien auf eine Partition“ auswählen und bestätigen**

**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

**15. Schritt: „Ja“ auswählen, um die Änderungen auf die Festplatte zu schreiben, und auf Weiter klicken**

**16. Schritt: Die Option „Nein“ auswählen, da kein weiteres Medium eingelesen werden soll und bestätigen**

**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

**17. Schritt: Land auswählen, in dem der Spiegelserver stehen soll**

**18. Schritt: Den Spiegelserver wählen, der die beste Internetverbindung herstellen kann**

**Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung**

**19. Schritt: Proxy-Daten leer lassen und auf Weiter klicken**

**20. Schritt: An der Paketverwendungserfassung nicht teilnehmen und mit „Nein“ fortfahren**

**Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung**

**Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung**

**21. Schritt: Auswahl der Softwarepakete, die zusätzlich installiert werden sollen und mit Weiter bestätigen**

**23. Schritt: Die angegebene Festplatte auswählen und auf Weiter klicken**

**24. Schritt: Die Installation ist abgeschlossen und mit Weiter fortfahren**

**22. Schritt: Mit „Ja“ den GRUB-Bootloader installieren und fortsetzen**

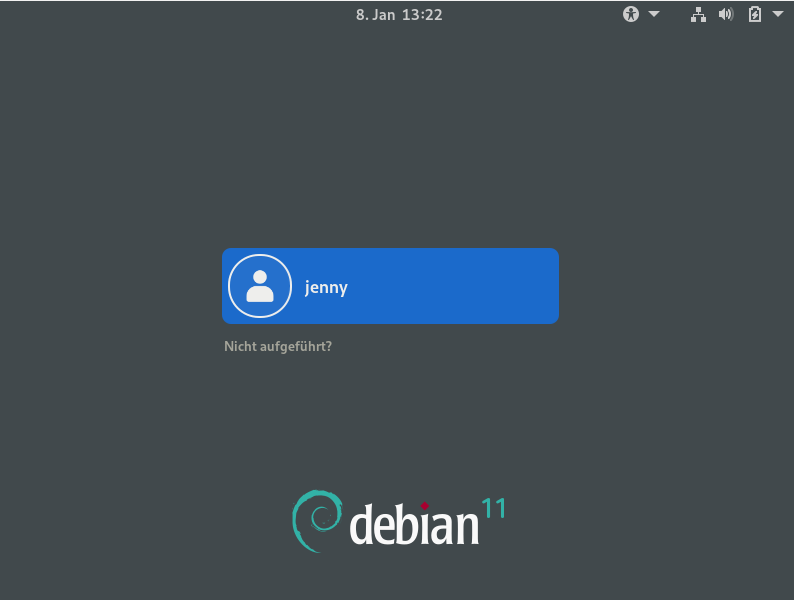
**Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung**

**25. Schritt: Dienste und Dateien werden geprüft und gestartet**

**24. Schritt: Debian 11 zum ersten Mal booten**

****

**26. Schritt: Anmeldebildschirm, um mit einem Benutzer zu starten**

**2. Installation von Docker:**

Zunächst laden wir Docker auf den Virtuellen Maschinen runter. Der Prozess ist auf allen vier Maschinen ist derselbe.

Damit wir Docker installieren können müssen wir ein Verzeichnis für Docker auf unseren Maschinen einrichten.

**Schritt 1:** Pakete aktualisieren und dann neue installieren, damit *apt* das Verzeichnis

über HTTPS benutzen kann

$ *sudo apt-get update*

$ *sudo apt-get install \*

*ca-certificates \*

*curl \*

*gnupg \*

*lsb-release*

**Schritt 2:** Docker GPG key hinzufügen

$ *sudo mkdir -p /etc/apt/keyrings*

$ *curl -fsSL https://download.docker.com/linux/debian/gpg | sudo*

*gpg --dearmor -o /etc/apt/keyrings/docker.gpg*

**Schritt 3:** Verzeichnis anlegen

$ *echo \*

*"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]*

*https://download.docker.com/linux/debian \*

*$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >*

*/dev/null*

Nachdem wir das Verzeichnis eingerichtet haben, installieren wir jetzt die Docker Engine.

**Schritt 1:** apt Pakete aktualisieren

$ *sudo apt-get update*

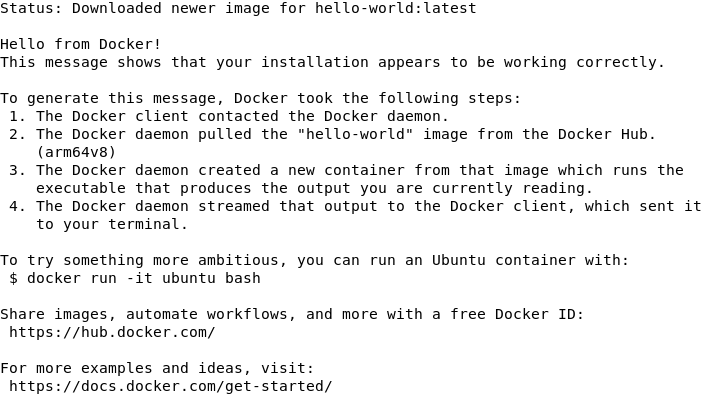
**Schritt 2:** die neuste Version der Docker Engine herunterladen

$ *sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-*

*plugin*

**Schritt 3:** testen mit*hello-world* image

$ *sudo docker run hello-world*



Bei erfolgreichem Download sollte die Ausgabe so ausschauen:

**Docker als non root-user benutzen:**

Vor jedem Docker Befehl müssen wir *sudo* anhangen, weil der Docker Deamon immer als *root* User läuft. Um das zu vermeiden, tun wir unseren User in der Docker Gruppe rein.

**Schritt 1:** Gruppe erstellen (falls sie noch nicht existiert)

$ *sudo groupadd docker*

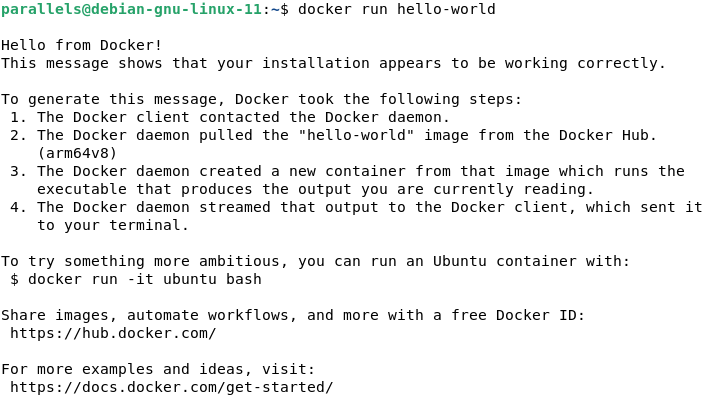
**Schritt 2:** User in der Docker Gruppe hinzufügen

$ *sudo usermod -aG docker $USER*

Danach sollte man seine VM neu starten, damit die Veränderungen gültig werden.

**Schritt 3:** *hello-world* image starten ohne sudo

$ *docker run hello-world*



Jetzt können wir ohne sudo docker benutzen, was uns später vieles erleichtert.

**3. NFS**

Wir benutzen *Network File System* als unser gemeinsamen Speichermittel für unseren Swarm Cluster. Dieses Protocol benutzt das Server/Client Prinzip. Eine Maschine ist der Server (Manager) und die anderen sind die Clients (Worker). Dabei sind, wie man schon ahnt, alle Daten auf dem Server gespeichert und die Clients können darauf zugreifen, so als wären die Dateien lokal gespeichert.

Dabei müsst ihr darauf achten, dass die Einrichtung von NFS auf der Server-Maschine anders ist als bei den Client-Maschinen.

**Schritt 1:** Pakete installieren

auf der Server-Maschine

$ *sudo apt update*

$ *sudo apt-get install nfs-kernel-server*

auf den Client-Maschinen

$ *sudo apt update*

$ *sudo apt-get install nfs-common*

**Schritt 2:** Verzeichnis erstellen und Rechte ändern

auf der Server-Maschine

$ *mkdir nfs-share*

$ *sudo chown nobody:nogroup nfs-share*

auf den Client-Maschinen

$ *mkdir nfs-server*

**Schritt 3:** NFS freigeben

auf der Server-Maschine

$ *sudo nano /etc/exports*

Graphical user interface, text

Description automatically generated

Nachdem öffnen von */etc/exports* sollte euer Bildschirm so ausschauen. Die drinstehenden Kommentare zeigen uns wie die Eingabe erfolgen soll.

Wir ergänzen ganz unten:

*/home/parallels/nfs-share 10.211.55.13(rw,sync,no\_root\_squash,no\_subtree\_check)*

Shareoptionen

Client\_ip

Pfad zum Verzeichnis auf Host

**Wichtig: kein Leerzeichen nach dem Komma machen!!!**

****

Der Pfad, der am Anfang steht */home/parallesl/nfs-share ist der Pfad zu dem geteilten Verzeichnis. Danach kommt die IP-Addresse oder Hostname des Clients, im unserem Fall 10.211.55.13. Die im klammer stehende Syntaxen sind die Shareoptionen.*

**rw:** *client kann Lesen und Schreiben*

**sync:** *Synchroner Datentransfer*

**no\_root\_squash:** *begrenzt die Rechte für einen SU per Remotezugriff*

**no\_subtree\_check:** *überprüft Unterverzeichnisbäumen nicht ab*

Wenn man mehrere Clients hat, so fügt man dasselbe wie oben aber mit der IP-Addresse des anderen Clients hinten an der Zeile an.

**Graphical user interface, text

Description automatically generated**

Nachdem wir die Datei gespeichert bzw. geschlossen haben, müssen wir den neuen Befehlen in der *exports*-Datei aktivieren.

auf der Server-Maschine

$ *sudo exportfs -ra*

Wir können uns die aktiven Maschinen auflisten

auf der Server-Maschine

$ *sudo exportfs -v*

Graphical user interface, text, application, email

Description automatically generatedAusgaben bei unserem Fall schaut so aus:

**(Schritt 4:** NFS Serverneustarten ) noch testen

auf der Server-Maschine

$ *sudo systemctl restart nfs-kernel-server*

**Schritt 5:** Firewall ändern

(mussen wir noch testen)

**Schritt 6:** Verzeichnisse *„mounten“*

auf den Client-Maschinen

Pfad zum Verzeichnis auf client

Pfad zum Verzeichnis auf Host

host\_ip

$ *sudo mount 10.211.55.15:/home/parallels/nfs-share /nfs-server*

Um es zu testen erstellen wir eine Datei mit *touch*:

auf den Client-Maschinen

$ *sudo touch nfs-server/testdatei*

Und wir lassen uns auf der Server-Maschine die Datei anzeigen:

Graphical user interface, text, application

Description automatically generatedauf der Server-Maschin

$ *ls -lh nfs-share/*

Normalerweise müss man den mount-Befehl jedes mal beim Neustarten der Maschine wieder eingeben, um das zu vermeiden schreiben wir was in der „fstab-Datei“ der Client-Maschinen rein.

auf den Client-Maschinen

$ *sudo nano /etc/fstab*

Graphical user interface

Description automatically generatedSo sollte bei euch der Editor ausschauen:

In der letzten Zeile schreiben wir:

10.211.55.15:/home/parallels/nfs-share /home/parallels/nfs-server nfs rsize=8192,wsize=8192 0 0

host\_ip

Dateisysteme checks

Systemsicherung

Optionen für die Lese- und Schreibegeschwindigkeit

Dateisystem

Pfad zum Verzeichnis auf Host

Pfad zum Verzeichnis auf Client

Graphical user interface

Description automatically generated

Jetzt müssen wir das Verzeichnis noch *„unmounten“*:

auf den Client-Maschinen

$ *sudo umount nfs-server*

Dann starten wir unsere Maschine neu und sehen, dass das NFS beim Start schon funktioniert.

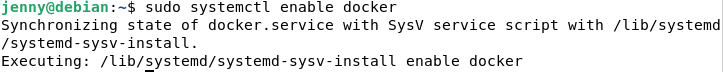
Graphical user interface, text, application

Description automatically generatedDiesen Prozess müssen wir auf jeder Client Maschine ausführen, damit alle ein gemeinsamen Speicher haben.

**4. Portainer installieren und einrichten:**

Schritt 1: Docker aktivieren

$ *sudo* systemctl enable docker

****

Den Docker Befehl herziehen

$ *sudo* docker pull portainer/portainer-ce

**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

Einen Platzhalter für die Daten erstellen

# docker volume create portainer\_data

****

Der Pfad importiert den Platzhalter, der für den oberen mit volume erstellt wurde

# docker run -d -p 8000:8000 -p 9000:9000 –-name=portainer –-restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer\_data:/date portainer/portainer-ce

****

**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

Als nächstes wird das YML-Manifest des Stapels abgerufen:

# curl -L https://downloads.portainer.io/ce2-16/portainer-agent-stack.yml -o portainer-agent-stack.yml

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Den heruntergeladenen YML-Manifest verwenden, um den Stack bereitzustellen:

# docker stack deploy -c portainer-agent-stack.yml portainer

Wir können überprüfen, ob die Portainer Server und Agent Container gestartet sind, indem wir docker ps ausführen

# docker ps

**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

Jetzt, wo die Installation abgeschlossen ist, können wir uns in unseren Portainer Server Instanz einloggen, indem wir einen Webbrowser öffnen und die IP-Adresse mit dem zugehörigen Port eingeben:

https://localhost:9000

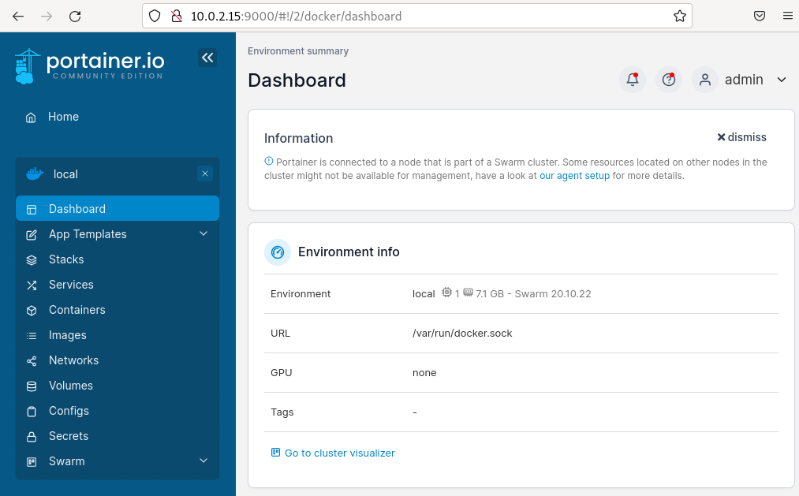
**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

Die Option Docker auswählen und mit Connect bestätigen

**Ein Bild, das Text enthält.

Automatisch generierte Beschreibung**

Nachdem alles bestätigt wurde, wählen wir den local aus, um in das Dashboard zu gelangen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**5. Docker Swarm einrichten**

Nachdem wir den Portainer installiert und eingerichtet haben, können wir einen Swarm erstellen. Man muss sicherstellen, dass der Docker-Engine-Daemon auf dem Host-Rechner gestartet ist.

**Manager Node einrichten:**

Schritt 1: Den Terminal öffnen,um sich per SSH auf dem Rechner einzuloggen, auf dem wir den Manager node ausführen möchten. In unserem Fall haben wir ein Rechner mit dem Namen manager verwendet.

docker swarm init --advertise-addr <MANAGER-IP>

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Mit dem Befehl --advertise-addr wird der Manager Node so konfiguriert, dass er seine Adresse als 10.0.2.15 veröffentlicht. Die anderen Nodes im Swarm müssen in der Lage sein, unter dieser IP-Adresse auf den Manager zuzugreifen.

Die Ausgabe enthält die Befehle zum Beitritt neuer Nodes zum Swarm. Die Knoten werden je nach dem Wert des --token-Flags als Manager oder Worker beitreten.

Schritt 3:

Den Befehl docker info ausführen, um den aktuellen Status des Swarms anzuzeigen:

$ docker info

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Schritt 4: Ausführung des Befehls docker node ls, um Informationen über die Nodes anzuzeigen:

$ docker node ls

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Der \* neben der Node-ID zeigt an, dass wir derzeit mit diesem Knoten verbunden sind.

**Worker Nodes einrichten:**

Sobald ein Swarm mit einem Manager Node erstellt wurde, können wir die Worker Nodes hinzufügen.

Schritt 1: Zunächst wird der Befehl ausgeführt, den die Ausgabe von docker swarm init vom Manager Node gegeben wurde, um einen Arbeitsknoten zu erstellen, der mit dem bestehenden Swarm verbunden wird:

$ docker swarm join \

--token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c \

10.0.2.15:2377

This node joined a swarm as a worker.

Wenn man den Befehl nicht zur Verfügung hat, dann kann folgender Befehl auf einem Manager Node ausführt werden, um den Join-Befehl für einen Worker abzurufen:

$ docker swarm join-token worker

To add a worker to this swarm, run the following command:

docker swarm join \

--token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c \

10.0.2.15:2377

Schritt 2: Den Terminal öffnen und mit dem Rechner verbinden, auf dem der Manager Node läuft und den Befehl docker node ls ausführen, um die Worker Nodes anzuzeigen:

$ docker node ls

Einleitung

Vorausetzungen

* Docker installieren
  + Docker ohne root laufen
* Portainer installieren

Vorgehensweise

- Probleme

- Lösungen

- Herausforderungen

- test

Zusammenfassung

* In echten leben

Quellen:

-<https://docs.docker.com/engine/swarm/swarm-tutorial/add-nodes/>

-<https://docs.docker.com/engine/swarm/swarm-tutorial/create-swarm/>

-[https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount- on-debian-11](https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-%20%20%20%20%20%20%20on-debian-11)

<https://www.howtoforge.com/tutorial/install-nfs-server-and-client-on-debian/>

Deckbild: <https://rocketloop.de/wp-content/uploads/2021/05/clustering-machine-learning-comprehensive-guide.jpg>