

## Inhalt der Checkliste

<b>Projektabgabe - Eigener Minecraft-Server.....</b>	<b>2</b>
1. Repository.....	2
Vorhandene Dateien.....	2
Dockerfile.....	2
docker-compose.yaml.....	2
README.md.....	3
2. Dokumentation.....	3
3. Hinweise.....	3
Sicherheitshinweise.....	3
Code-Konventionen.....	3
Testing.....	4

# Projektabgabe - Eigener Minecraft-Server

Bitte erfülle alle Punkte auf dieser Liste, bevor du das Projekt einreichst. Solltest du weitere Extras eingebaut haben, erwähne das kurz, damit sich die Mentoren dies bei Bedarf anschauen können.

## 1. Repository

### Vorhandene Dateien

- Es wurde eine .gitignore Datei angelegt, die alle irrelevanten Inhalte aus dem git repository ignoriert
- Es gibt eine docker-compose.yaml, die den Anforderungen des nächsten Punkts genügt
- Eine Datei namens README.md ist vorhanden und entsprechend der Kriterien unten erstellt worden
- Es befinden sich keine weiteren Dateien im Repository, ohne dass diese explizit in der README.md benannt und beschrieben werden.

### Dockerfile

- Es gibt ein Dockerfile in dem ein passendes Image zusammengestellt wird, um einen minecraft-server zu starten
  - siehe **3. Hinweise** für den Download-Link zur Server-Anwendung
- Im Dockerfile sollen alle notwendigen pakete im Basis-Image installiert und ggf. konfiguriert werden
- Stelle sicher, dass dein Server immer starten kann, für Umgebungs-Variablen solltest du evtl. Default-Werte verwenden
- Du solltest dabei kein vorgefertigtes Minecraft Image verwenden

### docker-compose.yaml

- Es gibt einen Service, der definiert und konfiguriert wird: mc-server
- Es gibt eine Env Konfiguration für den Minecraft-Server Service bei der alle unkritischen Variablen konfiguriert werden (keine Auth!)
- Es gibt notwendige Port-Freigaben, sodass der Container aus dem Internet erreichbar ist
- Es gibt Volumes-Konfigurationen für die Container Daten, sodass die Inhalte auf einem Dateisystem persistiert werden und der Spielstand nicht verloren geht

## README.md

- Die README sollte ein Inhaltsverzeichnis a.k.a. eine Table-of-Contents (ToC) enthalten
- Eine Sektion mit einer Beschreibung des Repositories muss vorhanden sein. In dieser Beschreibung sollte genannt werden was die wesentlichen Inhalte sind, was der Zweck des Repositories ist
- Eine Sektion "Quickstart" sollte als Teil der README enthalten sein. Hier sollten kurz Voraussetzungen genannt und eine Schnellstart-Anleitung beschrieben sein.
- Es ausführliche Variante der vorgenannten Sektion so als "Usage" enthalten sein. Hier soll genauer auf die Konfiguration und Konfigurierbarkeit eingegangen werden, d.h. es soll auch erklärt werden wie relevante Passagen modifiziert werden können, um andere Resultate zu erzielen.

## 2. Dokumentation

Die Dokumentation des Codes, sowie des Projektes soll im Repository in Form einer README Datei stattfinden.

Die Dokumentationssprache für alle Projekte (und zugehörige Unterlagen) ist englisch.

## 3. Hinweise

Du kannst dir eine Minecraft-Server Binary direkt auf der Minecraft Seite herunterladen unter (nutze hier die Java Version):

- <https://www.minecraft.net/de-de/download>.

### Allgemeine Hinweise

- Zusätzlich zu deinem GitHub Repository solltest du ein kurzes Loom Video (maximal 5min.) aufnehmen und bereitstellen, indem du kurz deine Abgabe zeigst und vorstellst was du getan hast - dabei musst du nicht alle Details erwähnen, jedoch sollst du auf alle relevanten Schritte kurz eingehen und diese zeigen

### Sicherheitshinweise

- Speichere keine SSH-Keys im Workspace deines Git-Repositories
- Speichere keine Passwörter, Tokens, oder Benutzernamen in deinem Code. Verwende hierfür stattdessen Environment-Variablen
- Speichere keine IP-Adressen, oder sonstigen sensiblen Informationen in einem Git Repository

### Code-Konventionen

- Für build-args, environment Variablen und Shell-Variablen gilt folgende Namenskonvention: UPPER\_CASE\_WITH\_UNDERSCORE

- Bei einer Referenz auf eine Variable sollte immer die {}-Notation verwendet werden um Fehler in der Interpretation zu vermeiden: `$(SOME_VAR_VALUE)`, statt: `$SOME_VAR_VALUE`
- Es sollten für build-args, oder Environment Variablen "Default"-Werte konfiguriert werden, allerdings nur dann wenn dies Sinn ergibt.
- Kritische Konfiguration wie Tokens, Passwörter oder ähnliches sollte nicht im Code-Repository gespeichert sein, sondern bspw. durch die Verwendung eines .env-files in einen Container hineingegeben werden

## Testing

Bevor du dein Projekt einreichst, solltest du die folgenden Dinge sichergestellt und getestet haben:

- Der Minecraft-Server ist erreichbar unter der IP-Adresse deiner Cloud-VM auf Port 8888
- Für das Testen hast du grundsätzlich zwei Wege: das Spiel starten und versuchen dich auf deinen Server zu verbinden (siehe nächster Punkt), oder mittels eines Skriptes versuchen eine Verbindung zum Minecraft Server aufzubauen. Dazu kannst du das folgende Python Modul verwenden: <https://github.com/py-mine/mcstatus>
- (optional) Du kannst dich von deinem Java-Minecraft Client auf deinem Computer mit dem Server auf deiner Cloud-VM verbinden und Minecraft spielen.
- Nach einem Neustart des Servers sind die konfigurierten Daten noch vorhanden und werden nicht gelöscht oder überschrieben. Dazu gehört unter anderem der Spielstand und die Konfiguration des Game-Servers
- Die Container der Services werden neugestartet, sobald ein Fehler passiert, der zum Terminieren des Containers führt.