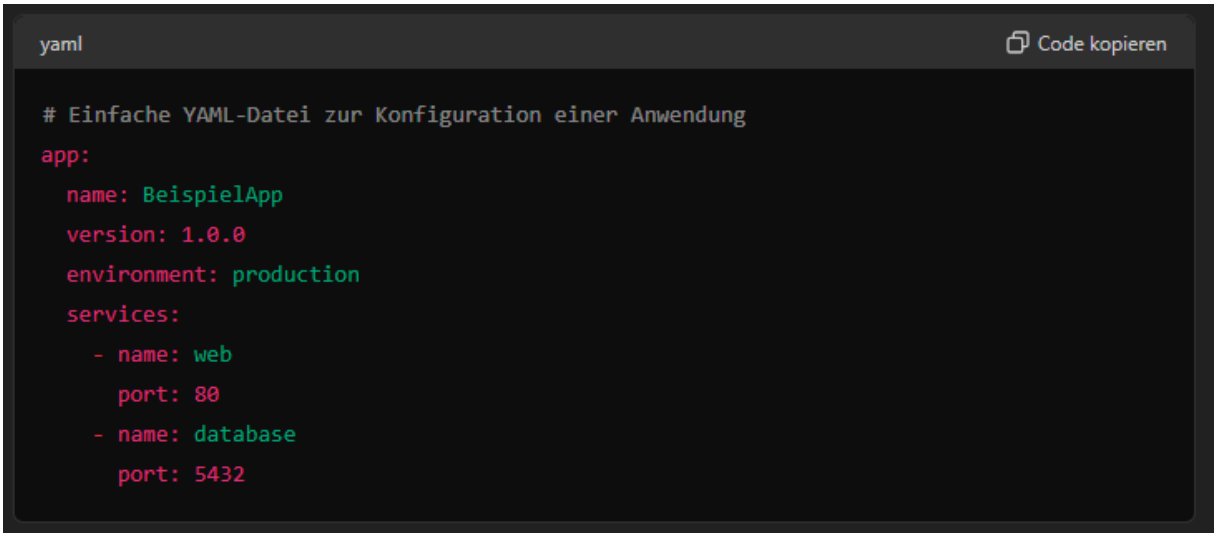
Zusammenfassung M347

**Lernziel 1: Sie können mit YAML-Dateien korrekt umgehen**

**Informationen:**

- YAML (YAML Ain't Markup Language) ist ein Datenserialisierungsstandard, der leserfreundlich und einfach strukturiert ist.
- YAML-Dateien werden häufig in Konfigurationsdateien und für die Konfiguration von DevOps-Tools verwendet.
- Syntaxregeln:
    - Einrückungen definieren die Hierarchie.
    - Listen werden mit Bindestrichen erstellt.
    - Schlüssel-Wert-Paare werden durch Doppelpunkte getrennt.




**Lernziel 2: Sie kennen die Syntax von docker-compose.yml und können die Schlüsselwörter korrekt anwenden**

**Informationen:**

- Docker Compose ist ein Tool zum Definieren und Verwalten von Multi-Container Docker-Anwendungen.
- Die docker-compose.yml-Datei enthält die Konfiguration für die verschiedenen Services.
- Wichtige Schlüsselwörter:
    - version: Spezifiziert die Version der Docker Compose-Datei.
    - services: Definiert die Services der Anwendung.
    - image: Gibt das Docker-Image an.
    - build: Definiert den Build-Kontext.

- ports: Mapped Ports zwischen Host und Container.
- volumes: Bindet Volumes zwischen Host und Container.

yaml Code kopieren

```yaml
version: '3.8'
services:
  web:
    image: nginx:latest
    ports:
      - "80:80"
  database:
    image: postgres:latest
    ports:
      - "5432:5432"
    environment:
      POSTGRES_PASSWORD: example
```

**Lernziel 3: Sie können das Kommando docker compose up/down mit den Schaltern -d und --build korrekt anwenden**

**Informationen:**

- docker compose up startet die in der docker-compose.yml definierten Container.
- -d: Startet die Container im Hintergrund (detached mode).
- --build: Erzwingt den Neuaufbau der Images vor dem Starten der Container.
- docker compose down: Stoppt und entfernt die Container sowie die Netzwerke, die von docker compose up erstellt wurden.

sh Code kopieren

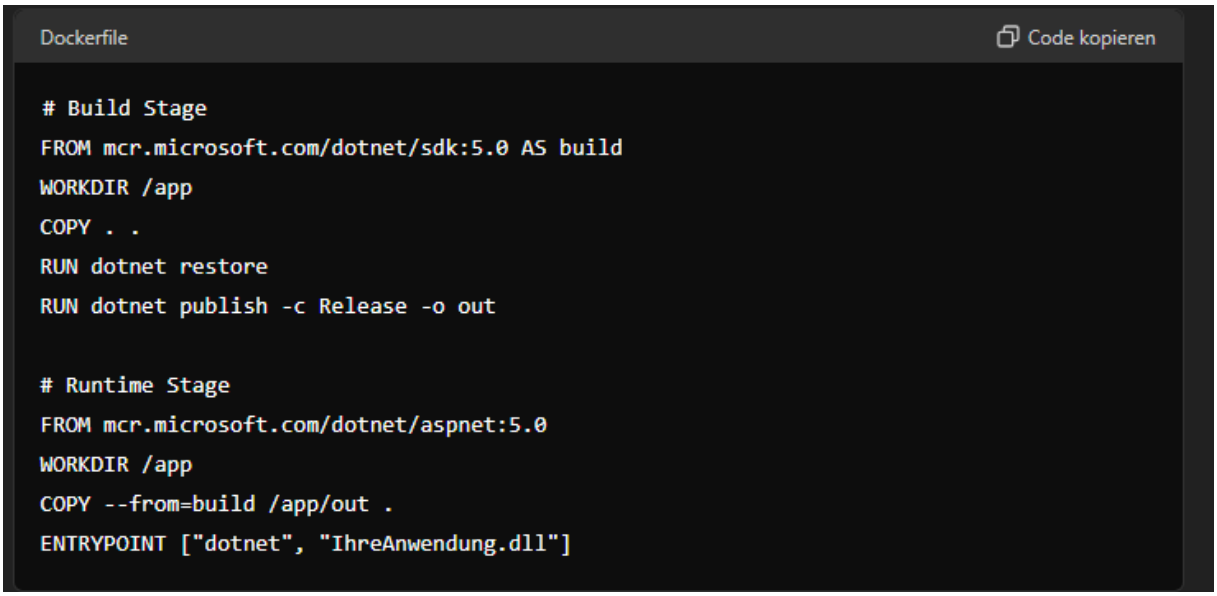
```sh
# Startet die Container im Hintergrund und baut die Images neu
docker compose up -d --build

# Stoppt und entfernt die Container
docker compose down
```

**Lernziel 5: Sie können Dockerfiles für Multistage Builds im Zusammenhang mit dotnet aufbauen**

**Informationen:**

- Multistage Builds reduzieren die Größe des finalen Images, indem sie unnötige Abhängigkeiten im finalen Build entfernen.
- Ein Dockerfile kann mehrere FROM Anweisungen enthalten, jede beginnt eine neue Stage.
- Im Zusammenhang mit .NET werden typischerweise zwei Stages verwendet: eine für den Build und eine für das Runtime-Image.



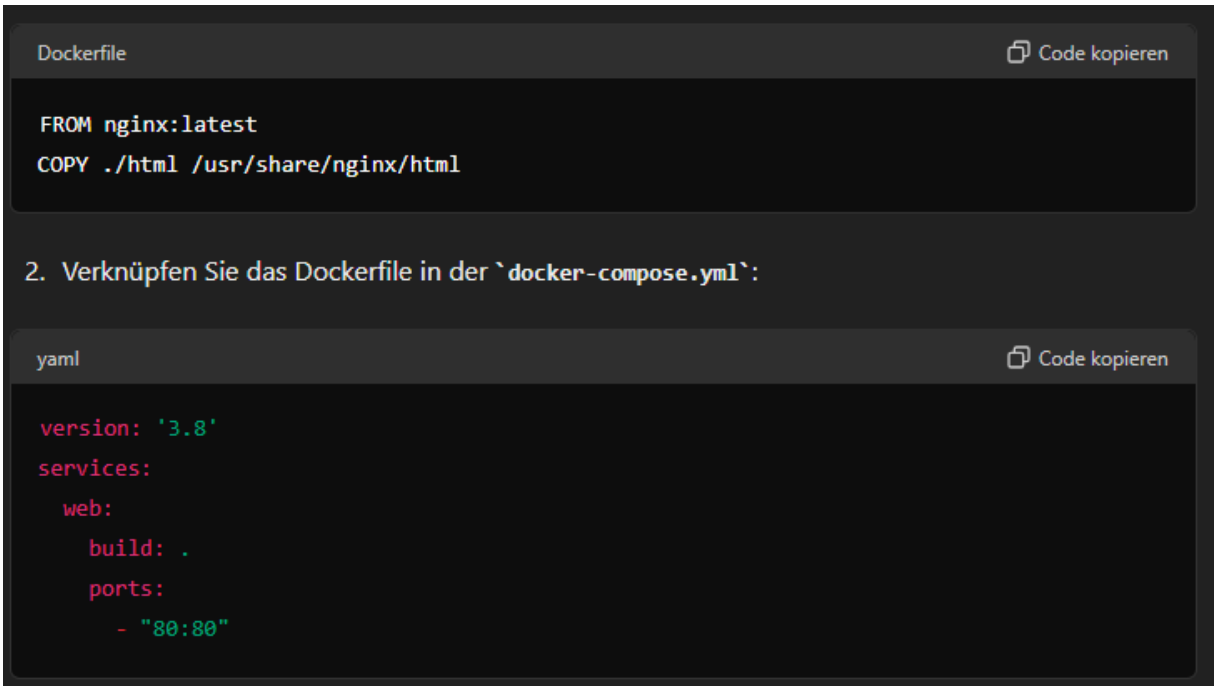

## Lernziel 6: Sie können Dockerfiles im Zusammenspiel mit docker compose korrekt verwenden

**Informationen:**

- Dockerfiles definieren das Image, das für einen Service in `docker-compose.yml` verwendet wird.
- Die `build` Direktive in `docker-compose.yml` kann auf das Verzeichnis mit dem Dockerfile verweisen.

**Beispiel:**

1. Erstellen Sie ein Dockerfile:



Dockerfile

Code kopieren

```
FROM nginx:latest
COPY ./html /usr/share/nginx/html
```

2. Verknüpfen Sie das Dockerfile in der `docker-compose.yml`:

yaml

Code kopieren

```
version: '3.8'
services:
  web:
    build: .
    ports:
      - "80:80"
```

**Lernziel 7: Sie können mit dotnet Web- und Konsolen-Anwendungen korrekt umgehen**

**Informationen:**

- .NET bietet Werkzeuge zur Erstellung von Web- und Konsolenanwendungen.
- Befehle wie dotnet new, dotnet run, und dotnet publish sind essenziell.
- Konsolenanwendungen sind einfache ausführbare Programme, während Webanwendungen auf ASP.NET Core basieren und in einem Webserver laufen.

**Beispiel:**

1. Erstellen und Ausführen einer Konsolenanwendung:

```
2. mkdir mynodewebApp
3. cd mynodewebApp
4. npm init
5. code .
6.
```

7. Erstellen und Ausführen einer Webanwendung:

**dotnet new web -o MeineWebApp**

**cd MeineWebApp**

**dotnet run**

# Die Webanwendung läuft jetzt auf http://localhost:5000

Secrets:

```
services:
   db:
     image: mysql:latest
     environment:
       MYSQL_ROOT_PASSWORD_FILE: /run/secrets/db_root_password
       MYSQL_PASSWORD_FILE: /run/secrets/db_password
       MYSQL_DATABASE: wordpress
       MYSQL_USER: wordpress
     secrets:
       - db_root_password
       - db_password
     ...
secrets:
   db_password:
     file: db_password.txt
   db_root_password:
     file: db_root_password.txt
```