**Docker Aufbau Gliederung**

**Wofür brauche ich Docker?**Das Hauptanwendungsgebiet von Docker ist recht simpel.   
Es dient dazu, eine Testumgebung für die zu entwickelnde Software einzurichten.   
Mithilfe des Docker-Image und dem Docker-File, wird die Umgebung „auch Docker-Container genannt“ im Windows Sub Linux erstellt.  
Dadurch steht dir dann beispielsweise eine Testdatenbank mit Testdaten zu Verfügung.   
Ein anderes Beispiel wären Rest-Endpoints, die du dann mit deiner selbstgeschriebenen App, dementsprechend ansteuern kannst.  
   
**Aber warum sollte ich Docker verwenden, wenn ich mir die Umgebung selbst einrichten kann?**Docker sorgt dafür, dass du dich nicht mehr mit unnötigen Installationen rumplagen musst.   
Du musst nur den Docker-Container Updaten und starten.

1. **Docker-File:** Das Docker-File ist für die Umgebung, die Beschreibung der Docker-Container und restlichen Ressourcen zuständig. Aus diesem Docker-File wird dann ein Docker-Image erzeugt, welches dann durch Docker gestartet werden kann.
2. **Docker-Image:** Das Docker-Image enthält alle im Docker-File beschriebenen Docker-Container und Ressourcen. Das Docker-Image wird aus dem Docker-File erzeugt.
3. **Docker-Container:** Den Docker-Container kann man mit einem virtualen Computer vergleichen. Dieser hat eine nach außen gerichtete Netzwerkverbindung, Programme, Bibliotheken und Treiber, die für den Betrieb eines Services benötigt werden.
4. **Docker-Compose:** Docker-Compose ist ein Kommandozeilen-Tool, welches dazu dient, mehrere Docker-Images über eine Docker-compose.yml Datei zu starten.   
   Damit kann man sich praktisch eine Startroutine für alle Container in einem Paket bauen.

Meine Eselsbrücke zu der ganzen Thematik ist einfach:   
Docker-File ist praktisch der Quellcode des Betriebsystems.   
Docker-Image ist praktisch das Betriebssystem Image.   
Docker-Container ist die gestartete und installierte Betriebssystem Instanz.