

Imperativ:

Imperative Konfiguration bedeutet, dass Sie Schritte angeben, die explizit ausgeführt werden müssen, um den gewünschten Zustand des Containers zu erreichen. Sie geben Anweisungen, wie der Container erstellt und verwaltet werden soll. Dies kann durch die Verwendung von Befehlen, Skripten oder Konfigurationsdateien erfolgen.

Vorteile	Nachteile
- Direkte Kontrolle über den Ablauf der Schritte.	- Geringere Portabilität, da die Konfiguration an bestimmte Umgebungen oder Plattformen gebunden sein kann.
- Feingranulare Anpassungsmöglichkeiten bei der Konfiguration.	- Schwierigkeiten bei der Wiederholbarkeit und Skalierbarkeit, insbesondere in komplexen Szenarien.

Deklarativ:

Bei der deklarativen Containerkonfiguration wird der gewünschte Zustand des Containers in einer Konfigurationsdatei festgelegt. Anstatt Schritte oder Befehle anzugeben, um den Container zu erstellen und zu konfigurieren, wird die Konfiguration in einer Datei definiert. Diese Datei wird dann an das Containerorchestrierungssystem wie Kubernetes übergeben, das den Container entsprechend der angegebenen Konfiguration erstellt und verwaltet. Wenn Änderungen vorgenommen werden müssen, wird die Konfigurationsdatei aktualisiert, und das Containerorchestrierungssystem sorgt dafür, dass der Container entsprechend aktualisiert wird.

Vorteile	Nachteile
- Portabilität, da die Beschreibung des Containers unabhängig von der zugrunde liegenden Plattform oder Umgebung ist.	- Möglicher Verlust der Kontrolle über den Ablauf der Schritte.
- Wiederholbarkeit und Skalierbarkeit, da die Konfiguration leicht auf andere Umgebungen angewendet oder dupliziert werden kann.	- Einschränkungen bei der Feinabstimmung der Konfiguration, da nicht alle spezifischen Schritte direkt angegeben werden können.
- Vereinfachte Verwaltung, da der Fokus auf dem gewünschten Zustand liegt und nicht auf den einzelnen Schritten.	

Was bedeutet die Abkürzung IaC geschrieben "Infrastructure as Code". Infrastructure as Code (IaC) bezieht sich auf den Ansatz, die Konfiguration, Bereitstellung und Verwaltung von IT-Infrastrukturressourcen mithilfe von Code zu automatisieren und zu standardisieren.

Mit welchem Protokoll arbeitet die REST-API Schnittstelle (Application Programming Interface)? Die REST-API-Schnittstelle (Representational State Transfer) arbeitet in der Regel mit dem Hypertext Transfer Protocol (HTTP). HTTP wird verwendet, um Anfragen an die API zu senden und die entsprechenden Antworten zu erhalten.

Erkläre, wie bei Docker Swarm die Verfügbarkeit sichergestellt wird. Bei Docker Swarm wird die Verfügbarkeit durch die Verteilung von Diensten auf mehrere Knoten im Cluster gewährleistet. Swarm ermöglicht die Skalierung von Diensten und die automatische Verteilung auf verfügbare Ressourcen. Wenn ein Knoten ausfällt, werden die Container automatisch auf andere Knoten im Cluster neu gestartet, um die Verfügbarkeit aufrechtzuerhalten.

Welche Vorteile bietet der Begriff Scalability? Skalierbarkeit bezieht sich auf die Fähigkeit eines Systems, mit steigender Arbeitslast umzugehen und sich an veränderte Anforderungen anzupassen. Docker Swarm bietet die Möglichkeit, Dienste horizontal zu skalieren, indem zusätzliche Instanzen eines Dienstes erstellt werden. Dadurch können mehr Ressourcen genutzt werden, um die Anwendungsleistung zu verbessern und eine bessere Lastverteilung zu erreichen.

Welche Aufgabe übernimmt der Scheduler bei Docker Swarm? Der Scheduler ist in Docker Swarm für die Zuweisung von Containern zu den Knoten im Cluster verantwortlich. Er analysiert die verfügbaren Ressourcen auf den einzelnen Knoten und entscheidet, auf welchem Knoten ein Container gestartet werden soll. Der Scheduler berücksichtigt dabei Faktoren wie die Auslastung der Knoten, die Ressourcenanforderungen der Container und eventuelle Platzierungsbeschränkungen, um eine optimale Verteilung zu erreichen.

Was bedeutet Split Brain und wie reagiert Docker Swarm auf eine solche Situation? Split Brain bezieht sich auf eine Situation, in der die Kommunikation zwischen den Knoten eines Clusters unterbrochen ist und die Knoten nicht mehr in der Lage sind, sich zu einigen, welcher Knoten die Kontrolle übernehmen soll. Dies kann zu inkonsistenten Zuständen führen, da verschiedene Knoten gleichzeitig als "Master" agieren können. Docker Swarm verwendet ein Konsensus-Algorithmus namens Raft, um Split Brain zu verhindern. Raft ermöglicht die Wahl eines eindeutigen Anführers (Leader), der die Entscheidungen im Cluster trifft. Wenn die Kommunikation wiederhergestellt wird, wird der Raft-Algorithmus verwendet, um die Knoten zu synchronisieren und den eindeutigen Anführer zu bestimmen.

Was ist die Aufgabe eines Loadbalancers? Die Aufgabe ist es, die Auslastung gleichmässig zu verteilen, so dass die Umgebung effizient genutzt werden kann.

Woran erkennt man, dass die vorangegangene Übung als Imperativ- und diese Übung als Deklarativ bezeichnet wird?

Beim Deklarativen wird alles im docker-compose.yml definiert. Imperativen, werden die Dinge nachträglich auf der Kommandozeile ausgeführt.

Was bedeutet der Begriff Desired State Der Desired State, ist in diesem Fall die Anzahl und Zustand der Container, die am laufen sein sollten. Im allgemeinen kann man sagen, dass es ein Zustand ist, in dem die Umgebung sein sollte. Gesetzt würde dieser, beim Erstellen des Services mit dem teil -- replica 5. Anschliessend haben wir diesen erhöht (docker servcie scale web=8).

Welches File bestimmt, wie die Microservices (Container) zusammenarbeiten? - Das wird im Docker-compose.yaml definiert

Was ist ein Redis-Cache? Ein Redis-Cache ist ein schneller Zwischenspeicher, der häufig abgerufene Daten im Arbeitsspeicher speichert, um die Leistung von Anwendungen zu verbessern. Es ermöglicht schnellen Zugriff auf diese Daten, ohne auf die eigentliche Datenquelle zugreifen zu müssen.

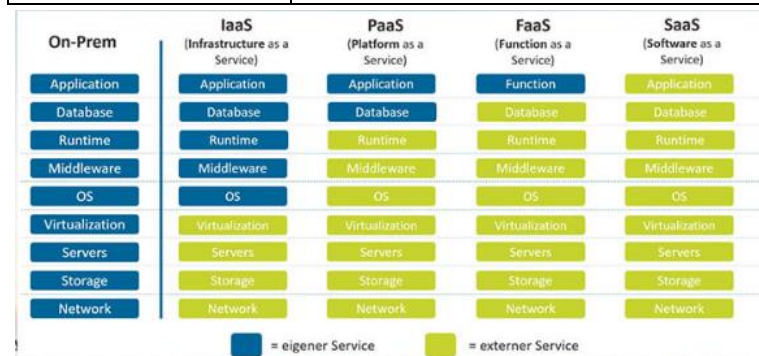
Auf welchem Port horcht Redis standardmässig und wo wird dieser Port definiert? - Normalerweise wird der Port 6379 verwendet, dieser kann im Docker-Compose angepasst werden

Erkläre, welcher Eintrag im Config-File docker-compose.yml im Vergleich zur letzten Übung dazugekommen ist - Es ist der deploy teil hinzugekommen, da werden die Anzahl replicas definiert

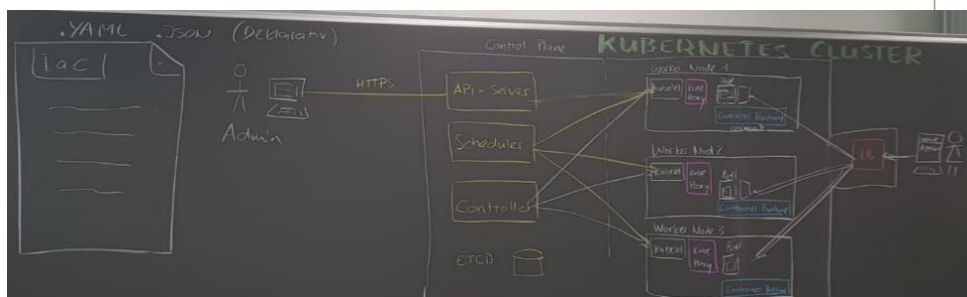
Tool	Funktion	Verwendung
Docker	Eine Plattform zur Containerisierung von Anwendungen.	Einzelne Container lokal erstellen und ausführen.
Docker Compose	Ein Tool zur Definition und Ausführung von Multi-Container-Anwendungen.	Lokale Anwendungen mit mehreren Containern definieren und ausführen.
Docker Swarm	Ein Container-Orchestrierungstool, um einen Cluster von Docker-Hosts zu einem virtuellen Host zu vereinen.	Cluster von Docker-Hosts erstellen und verwalten.
Docker Stack	Ein Tool zur Bereitstellung von Anwendungen auf einem Docker Swarm-Cluster.	Anwendungen mithilfe einer Compose-Datei auf einem Swarm-Cluster bereitstellen und verwalten.

Command	Erklärung
docker-compose up -d	Nun kann man mit diesem Befehl den Docker starten
docker-compose down	Mit diesem Befehl kan man den Docker wieder herunterfahren
docker swarm init	Docker-Swarm-Manager zu initialisieren und einen neuen Docker Swarm-Cluster zu erstellen.
docker service scale web=10	Container skalieren (hier auf 10)
docker service create --name swarm-web -p 8090:8080 --replicas 5 registry.gitlab.com/ser-cal/docker-bootstrap/webserver_one:1.0	Docker erstellen mit swarm. (Imperativ)
docker stack deploy -c docker-compose.yml zaehler	Docker erstellen mit stack. (Deklarativ)
docker stack ls	Zeigt Name, Services und Orchestrator (Swarm) an
docker stack services zaehler	Zeigt die beiden Container an - dabei wurde der Container "counter_web_fe" 10x repliziert
docker stack ps zaehler	Zeigt jeden einzelnen Container
docker stack rm zaehler	Alle Conatiner mit zaehler löschen

Bauen und starten des Images	<p>Docker Image erstellen: docker image build -t registry.gitlab.com/ck2005/m346-kaelin/webserver_one:1.0 .</p> <p>Docker image auf das Gitlab registry pushen: docker push registry.gitlab.com/ck2005/m346-kaelin/webserver_one:1.0</p> <p>Docker Container starten: -- name steht für den Namen, -p steht für den Port, der erste ist der, der von aussen erreichbar ist. docker container run -d --name container-webserver -p 8080:8080 registry.gitlab.com/ck2005/m346-kaelin/webserver_one:1.0</p> <p>Starten des docker-compose.yml: docker-compose up -d</p> <p>Starten des Services: docker stack deploy -c docker-compose.yml Service Name</p>
Anzeigen lassen	<p>Alle Container anzeigen lassen: mit grep -i und dem Namen kann man sich bestimmte Container anzeigen lassen docker container ls -a grep -i</p> <p>Container von dem docker-compose anzeigen lassen: docker container ls -a grep -i mulicontainer</p> <p>Image anzeigen lassen: docker image ls grep -i mulicontainer</p> <p>ID's der Container eines Services anzeigen lassen: docker ps grep "service/swarm name" Löschen bestimmter Container: docker container rm -f</p>
Stoppen und löschen des Images	<p>Container stoppen: docker container stop</p> <p>Container löschen: docker Container rm</p> <p>Image löschen: docker image rm</p> <p>Docker-compose Container löschen: docker-compose down</p> <p>Löschen Docker swarm docker stack rm</p> <p>Löscht den Service und alle Container vond dem Service: docker stack rm zaehler</p>



Aspekt	Docker Swarm	Kubernetes
Skalierbarkeit	Begrenzte Skalierbarkeit für sehr große Workloads.	Bietet hohe Skalierbarkeit und ist gut für große Workloads geeignet.
Anpassungsmöglichkeiten	Unterstützt weniger Plugins und Add-Ons von Drittanbietern.	Bietet umfangreiche Unterstützung für Plugins und Add-Ons von Drittanbietern.
Ökosystem	Hat ein kleineres Ökosystem mit weniger Tools und Diensten.	Verfügt über ein umfangreiches Ökosystem mit vielen Tools und Diensten.
Flexibilität	Weniger flexibel mit begrenzten Konfigurationsmöglichkeiten.	Sehr flexibel mit umfangreichen Konfigurationsmöglichkeiten für fortgeschrittene Anwendungen.
Komplexität	Einfacher zu bedienen, aber kann für größere Anwendungen komplex sein.	Komplexer, aber gut geeignet für größere und komplexere Anwendungen.
Funktionen	Bietet nicht alle Funktionen und Fähigkeiten von Kubernetes.	Verfügt über umfangreiche Funktionen und Fähigkeiten wie die Verwaltung mehrerer Cluster.
Unterstützung	Wird von Docker Inc. entwickelt und unterstützt.	Wird von der Cloud Native Computing Foundation (CNCF) unterstützt und hat breitere Unterstützung.



Declarative vs. Imperative

• Beschreibe den erwünschten Zustand deiner Applikation in einem Config file (vorzugsweise YAML oder JSON-File).

• Übergebe das Config file mit einem einzigen Befehl via REST-API an den entsprechenden Service und überlasse ihm den Aufbau der Infrastruktur

• In der Fachsprache «Infrastructure as Code» genannt

Vorteile:

• Dokumentation ist im Code eingebettet

• Konsistenter Aufbau der Plattform (immer gleich)

• Plattform kann dynamisch angepasst / skaliert werden