TECH STARTER

Arbeitsblatt: Einführung in Kubernetes-Objekte

Grundlegende Objekte – Pods

Was ist ein Pod in Kubernetes?
→ Kleinste deploybare Einheit, die einen oder mehrere Container enthält.

Warum benötigt Kubernetes Pods?
→ Gruppiert Container als eine logische Einheit mit gemeinsamem Netzwerk & Speicher.

Was läuft innerhalb eines Pods?
→ Einer oder mehrere Container + Storage + Netzwerkeinstellungen + Metadaten.

Kann ein Pod mehrere Container enthalten? Warum oder warum nicht?
→ Ja, wenn sie eng zusammenarbeiten (z. B. Sidecar-Pattern).

Was passiert mit einem Pod, wenn er abstürzt?
→ Kubernetes kann ihn neustarten oder durch einen neuen ersetzen.

Wie lange lebt ein Pod?

→ Bis er gelöscht oder ersetzt wird (Pods sind nicht dauerhaft).

Grundlegende Objekte – ReplicaSet

Was ist ein ReplicaSet?
→ Stellt sicher, dass eine festgelegte Anzahl an Pods läuft.

Welches Problem löst ein ReplicaSet in einem Cluster?
→ Automatische Wiederherstellung von abgestürzten oder gelöschten Pods.

Wie stellt ein ReplicaSet sicher, dass immer eine bestimmte Anzahl von Pods läuft?
→ Kubernetes prüft kontinuierlich die gewünschte vs. tatsächliche Anzahl.

Was passiert, wenn ein Pod in einem ReplicaSet gelöscht wird?
→ Ein neuer Pod wird automatisch erstellt.

Kann man ein ReplicaSet alleine nutzen oder wird es meistens durch etwas anderes verwaltet?

→ Meistens durch ein **Deployment** verwaltet.

Grundlegende Objekte – Deployment

Was ist ein Deployment?
→ Verwaltungsschicht für ReplicaSets, ermöglicht Updates und Rollbacks.

Warum sollte man Deployments statt direkt ReplicaSets verwenden?
→ Vereinfachte Verwaltung, Updates, Skalierung & Rollbacks.

Wie hilft ein Deployment bei der Aktualisierung (Rolling Update) von Anwendungen?
→ Erstellt neue Pods mit neuer Version & löscht alte schrittweise.

Welche Vorteile hat ein Deployment gegenüber einem direkten ReplicaSet?

→ **Versionskontrolle, automatisierte Updates, einfache Skalierung.**

Unterschiede & Verbindungen

Was ist der Unterschied zwischen einem Pod und einem Deployment?
→ **Pod:** einzelne Einheit, **Deployment:** verwaltet mehrere Pods über ReplicaSets.

Was ist der Unterschied zwischen einem Deployment und einem ReplicaSet?
→ **Deployment verwaltet ReplicaSets**, die wiederum Pods verwalten.

Was ist der Zusammenhang zwischen Deployment, ReplicaSet und Pod?
→ Deployment → erstellt ReplicaSet → erzeugt & verwaltet Pods.

Wie erstellt ein Deployment neue Pods?
→ Durch ein ReplicaSet, das die gewünschte Anzahl sicherstellt.

Wie überwacht ein Deployment, ob alle Pods korrekt laufen?

→ Durch regelmäßige Statusprüfungen und Neustarts.

Netzwerk & Kommunikation – Services

Was ist ein Kubernetes Service?
→ Abstraktion, um Pods über eine feste IP-Adresse erreichbar zu machen.

Warum brauchen wir einen Service, wenn Pods ohnehin eine IP-Adresse haben?
→ Pods sind kurzlebig, ihre IP ändert sich ständig.

Wie funktioniert ein Service mit mehreren Pods (z. B. bei Skalierung)?
→ Lastverteilung über alle zugehörigen Pods (Load Balancing).

Welche Service-Typen gibt es in Kubernetes (ClusterIP, NodePort, LoadBalancer)?
→ **ClusterIP, NodePort, LoadBalancer, ExternalName**.

Wie findet ein Service die richtigen Pods (z. B. über Labels)?

→ Über **Labels & Label-Selectoren**.

TECH STARTER

Verbindungen zwischen Objekten

Was ist der Zusammenhang zwischen einem Service und einem Deployment?
→ Service macht die vom Deployment verwalteten Pods erreichbar.

Wie kommuniziert ein externer Client mit einem Pod über einen Service?
→ Über einen **NodePort** oder **LoadBalancer**.

Was ist der Unterschied zwischen einem ClusterIP-Service und einem NodePort-Service?

→ **ClusterIP:** Nur innerhalb des Clusters erreichbar, **NodePort:** Externer Zugriff über jeden Node.

Skalierung & Anpassung – HPA

Was ist ein Horizontal Pod Autoscaler (HPA)?
→ Automatische Skalierung von Pods basierend auf Last.

Wie funktioniert die automatische Skalierung mit HPA?
→ Überwacht Metriken & passt die Pod-Anzahl dynamisch an.

Welche Metriken werden zur Skalierung verwendet?
→ **CPU, Speicher, benutzerdefinierte Metriken.**

Kann man HPA mit einem Deployment verwenden?
→ Ja, HPA passt die Anzahl der Pods in einem Deployment an.

Was passiert, wenn die CPU-Auslastung eines Pods steigt?

→ HPA erstellt neue Pods, um die Last zu verteilen.

TECH STARTER

Unterschiede & Verbindungen zur Skalierung

Was ist der Unterschied zwischen einem ReplicaSet und einem Horizontal Pod Autoscaler?
→ **ReplicaSet hält feste Anzahl von Pods**, HPA passt sie dynamisch an.

Wie arbeiten Deployment, ReplicaSet und HPA zusammen?
→ **Deployment verwaltet ReplicaSet, HPA ändert die Anzahl der Pods.**

Was passiert, wenn man manuell die Anzahl der Pods ändert und gleichzeitig ein HPA aktiv ist?

→ HPA überschreibt manuelle Änderungen bei der nächsten Skalierungsprüfung.

Weitere wichtige Objekte – ConfigMap, Secret, Namespace

Was ist eine ConfigMap und wofür wird sie verwendet?
→ Speichert **nicht-sensible Konfigurationsdaten** (z. B. Umgebungsvariablen).

Wie kann eine Anwendung auf Daten in einer ConfigMap zugreifen?
→ Als **Umgebungsvariable, Kommandozeilenargument oder Volume**.

Was ist der Vorteil gegenüber hardcodierten Konfigurationswerten?
→ **Flexibilität & Trennung von Code und Konfiguration.**

Was ist ein Secret in Kubernetes?
→ Speichert **sensible Daten (Passwörter, Tokens, Zertifikate).**

Wie unterscheidet sich ein Secret von einer ConfigMap?
→ **Secrets sind verschlüsselt gespeichert, ConfigMaps nicht.**

Wie schützt Kubernetes die Inhalte eines Secrets?
→ **Base64-Verschlüsselung & RBAC (Role-Based Access Control).**

Was ist ein Namespace?
→ **Virtuelle Trennung innerhalb eines Clusters** zur Organisation von Ressourcen.

Warum sind Namespaces in größeren Clustern wichtig?
→ **Ermöglichen Isolation & bessere Ressourcennutzung.**

Können zwei Deployments denselben Namen haben, wenn sie in verschiedenen Namespaces sind?

→ **Ja, da Namespaces eine logische Trennung bieten.**