

Handlungsziele

Modul 169 Services mit Containern bereitstellen

✓ Definiert die erforderliche Umgebung für die automatisierte Bereitstellung von Services. ✓

1. Kennt die Grundprinzipien der monolithischen und der Microservice-Architektur.
2. Kennt die wesentlichen Vor- und Nachteile unterschiedlicher Architekturen und deren Nutzen auf Serviceebene aufzeigen (z.B. Microservice-Architektur).
3. Kennt die Vorgehensweise, wie Services in Container verpackt werden.
4. Kennt die Vorgehensweise, wie Services im Backend bereitgestellt und wie die Services im Frontend von Clients genutzt werden.
5. Kennt die einer Containerumgebung zugrundeliegende Architektur (Daemon, Client/Server, Images, Container, Registry).
6. Kennt die Abhängigkeiten von Services und deren Bereitstellung in einer lokalen Infrastruktur (Beispiele: persistente Datenspeicherung, Vernetzung und andere).

✓ Dokumentiert den logischen und physischen Aufbau der Umgebung in einem Netzwerkschema mit servicespezifischen Angaben. ✓

1. Kennt die erforderliche Dokumentation der Services mit den Standard-Netzwerksymbolen und der Unterscheidung, zwischen logischer und physischer Architektur.
2. Kennt die Netzwerkkonfiguration der Systeme (IP-Adressierung, Subnetzmaske, Standardgateway, DNS, Serviceport-Angaben) und deren Abhängigkeiten.
3. Kennt die unterschiedlichen Darstellungsarten, welche für Dokumentation verwendet werden können (Blockschaltbild, logischer Netzwerkplan, Schichtenmodell für Kommunikation und Systeme) und setzt situativ die sinnvollste Darstellungsart ein.

✓ **Erstellt und dokumentiert den für die Service-Breitstellung erforderlichen Code versioniert.** ✓

1. Kennt Tools mit Syntaxunterstützung für die Code Erstellung.
2. Kennt den Nutzen und Einsatzzweck von Versionsverwaltungssystemen und setzt diese situativ korrekt ein, sodass der erstellte Code versioniert und kommentiert verfügbar ist.
3. Kennt die Vorgehensweise, wie Systeme und Services codebasiert (Infrastructure as Code - IaC) aufgebaut werden.
4. Kennt den Lebenszyklus (Erstellen – Editieren – Testen – Einsetzen – Verwerfen).
5. Kennt die Vorgehensweise, wie die erstellen Services getestet werden können.
6. Kennt Methoden, mit welchen die Codekonsistenz überprüft werden kann.

✓ **Plant und realisiert die servicespezifischen Sicherheitsanforderungen.** ✓

1. Kennt die Sicherheitsmassnahmen und Methoden, um einen sicheren Betrieb der Containerumgebung zu gewährleisten.
2. Kennt die Vorgehensweise, wie definierten Sicherheitsmassnahmen in die Dokumentation transferiert und dargestellt werden kann.
3. Kennt Methoden, mit welchen die Sicherheitsmassnahmen auf die Services angewendet werden können.
4. Kennt die Wichtigkeit der Sicherheitsmassnahmen und überprüft die Wirksamkeit mit Hilfe eines Testkonzepts.

✓ **Erstellt die erforderlichen Datenverbindungen zwischen unterschiedlichen Services.** ✓

1. Kennt die erforderlichen Massnahmen, um die Datenverbindung zwischen Services herzustellen (Beispiel: Portfreigaben, Berechtigungen, Authentifizierung).
2. Kennt die Möglichkeiten des systemübergreifenden Datenaustauschs über definierte Schnittstellen (Beispiele: Synchrone und Asynchrone Kommunikation zwischen Services, Beispiele: REST Paradigma, Message Bus, API Gateway, Monitoring).
3. Kennt die Vorgehensweise und Zusammenhänge der zentralen und persistenten Datenspeicherung.

✓ **Stellt die Services in der definierten Umgebung reproduzierbar bereit.**



1. Kennt die Vorgehensweise, wie Container servicebezogen verwaltet werden (Starten, Funktionsfähigkeit überprüfen, Stoppen, Löschen).
2. Kennt die Möglichkeit, Images von Containern in Registries für die weitere Verwendung verfügbar zu machen.

✓ **Administriert und überwacht die bereitgestellten Services.**



1. Kennt die erforderlichen Massnahmen, um die Funktionalität der Services und den zugrundeliegenden Containern automatisiert zu überwachen.
2. Kennt die Möglichkeiten, wie die verfügbaren Ressourcen gemanagt (Beschränkung von Containern) werden können und definiert die nötigen Massnahmen mit Hilfe einer entsprechenden Berechnung aufgrund von konkreten Annahmen.

✓ **Versteht anhand der Dokumentation die Funktionalität der Services und unterstützt bei der Fehlersuche.**



1. Kennt die Zusammenhänge in einer containerbasierten Infrastruktur.
2. Kennt die Grundfunktionalität der Tools, welche die Fehlersuche in Containern unterstützen.