DEUTSCHE VERWALTUNGSCLOUD
EIN PRODUKT DES IT-PLANUNGSRATS

Detailstandard 10: Detailstandards der CI/CD-Pipeline

Version 1.0 (8.5.2025)

Inhaltsverzeichnis

1 Zusammenfassung

Dieses Dokument beschreibt im Kapitel Anforderungen die zu erfüllenden Anforderungen in diesem Standardisierungsbereich. Im Kapitel Standardisierung werden vorgegebene Architekturen und Realisierungen beschrieben. Diese stellen eine Konkretisierung der beschriebenen Anforderungen da.

2 Anforderung

Die Anforderungen lassen sich unterteilen in die folgenden Aspekte:

- Verwaltung der Codebase (Softwarelieferant)
- Qualitätssicherung der Sourcen (Softwarelieferant)
- Build-Erstellung (Softwarelieferant)
- Release und Deployment (Softwarelieferant, Softwarebetreiber)

3 Standardisierung

Grundsätzlich gilt das Whitepaper Standardisierung des Deployments als zentraler Orientierungspunkt für die Ausgestaltung der CI/CD Pipeline.

Die aus diesem Whitepaper übernommenen Vorgaben sind in den folgenden Tabellen mit einer ID WP_SD_... gekennzeichnet.

3.1 Vorgaben für Softwarelieferanten

3.1.1. Verwaltung der Codebase

Zur Sicherstellung der Nachvollziehbarkeit von Codeänderungen werden folgende Vorgaben definiert:

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_5.1_1	Software-lieferant	SOLL	die Source-Files und Configuration-Files in einem Git-Repository speichern.
WP_SD_5.1_2	Software-lieferant	SOLL	seine Codebase in einem Versionsmanagementsystem verwalten.
WP_SD_5.1_3	Software-lieferant	SOLL	Es soll immer eine eindeutige Korrelation zwischen einer Codebase und einer App bestehen. Jede Änderung ist zu versionieren.
WP_SD_5.1_4	Software-lieferant	MUSS	Das Staging muss in den Repositories berücksichtigt werden. Es muss für jede Stage die zugehörige Codebase bekannt sein.
WP_SD_5.1_5	Software-lieferant	MUSS	Für die Verwaltung des Codes ist ein geeigneter Workflow zu wählen und die Dokumentation für die Nutzenden zugänglich zu machen.
WP_SD_5.1_6	Software-lieferant	MUSS	Es müssen Git-Regeln wie z.B - Code und Konfigurationen trennen - Entwicklungs- und Produktion in unterschiedliche Directories trennen - keine Branches für unterschiedliche Environments verwenden festgelegt und durchgesetzt werden.
DS_10_A001	Software-lieferant	MUSS	für die Unveränderlichkeit seiner Releases Sorge tragen. Hierfür muss ein Softwarelieferant ein Code-Repository einsetzen und Releases hierin in einer unveränderlichen Art und Weise ablegen. Für das Code-Repository muss nach BSI-Vorgaben (OPS.1.1.2.A26) eine Backupstrategie eingerichtet werden.

3.1.2. Qualitätssicherung der Sourcen

Im Rahmen eines automatischen Deployments ist die Qualitätssicherung integraler Bestandteil des Deploymentprozesses.

Aus diesem Grund werden folgende Vorgaben empfohlen:

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_5.2_1	Software-lieferant	MUSS	Für die (automatisierte) Qualitätssicherung müssen Festlegungen im Projekt getroffen werden.
WP_SD_5.2_2	Software-lieferant	SOLL	Die Prüfung der Sourcen (Sicherheit und Qualität) sollte automatisch erfolgen.
WP_SD_5.2_3	Software-lieferant	SOLL	Das Einchecken und Mergen in Hauptbranches sollte nur bei erfolgreicher Prüfung nach der Ausführung von Unittests abgeschlossen werden können.
WP_SD_5.2_4	Software-lieferant	MUSS	Quelltext und Konfigurationsdaten müssen immer getrennt werden. Eine Anpassung der Konfiguration soll ausschließlich über Konfigurationsobjekte/Umgebungsvariablen usw. erfolgen.
WP_SD_5.2_5	Software-lieferant	MUSS	Es dürfen keine Credentials, private Schlüssel, Secrets usw. unverschlüsselt im Repository abgelegt werden.

Fortsetzung auf nächster Seite

Fortsetzung von vorheriger Seite

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_5.2_6	Software-lieferant	MUSS	Das Deployment muss so gestaltet werden, dass die Nutzung der abhängigen Dienste individuell konfiguriert werden kann
DS_10_A002	Software-lieferant	MUSS	dafür Sorge tragen, dass im Rahmen einer automatischen Bereitstellung die Qualitätssicherung integraler Bestandteil des Bereitstellungsprozesses ist. Innerhalb der Qualitätssicherung ist der Code via Tests und Statischer-Code-Analyse zu prüfen 0und ein SBOM zu generieren. Das generierte SBOM ist bei der Bereitstellung des Artefakts mit anzuhängen.

Folgende Punkte sollten explizit betrachtet werden:

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_5.2_7	Software-lieferant	SOLL	Alle Abhängigkeiten sollen explizit deklariert und isoliert werden.
WP_SD_5.2_8	Software-lieferant	SOLL	Die Prüfung auf Schwachstellen sollte bereits zu einem frühen Zeitpunkt erfolgen.
WP_SD_5.2_9	Software-lieferant	SOLL	Es soll sichergestellt werden, dass SBOMs (Open Source-Tool: Syft = CLI bzw. Go Library) erzeugt und verwendet werden.
WP_SD_5.2_10	Software-lieferant	MUSS	Im Source-Code dürfen keine ungenutzten Bibliotheken referenziert werden.

3.1.3. Build-Erstellung

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_5.3_1	Software-lieferant	MUSS	Es muss eine Trennung in Build-, Release- und Run-Phase erfolgen (siehe auch The Twelve-Factor App).
WP_SD_5.3_2	Software-lieferant	SOLL	Es soll eine Phase Test zwischen Build und Release verwendet werden.
WP_SD_5.3_3	Software-lieferant	MUSS	Jeder Build muss eine eindeutige ID haben.
WP_SD_5.3_4	Software-lieferant	MUSS	Eine Veränderung des Containers zur Laufzeit ist nicht zulässig. Alle Änderungen müssen über die Versionsverwaltung dokumentiert werden.
WP_SD_5.3_5	Software-lieferant	SOLL	Jede Änderung erzeugt eine neue Release. Jedes Release sollte eine eindeutige Release-ID und eine Versionsnummer haben.
WP_SD_5.3_6	Software-lieferant	MUSS	Jedes Release muss eindeutig auf die benutzten Komponentenversionen zurückgeführt werden können.
WP_SD_5.3_7	Software-lieferant	SOLL	Für die Versionsnummer soll Semantic Versioning 2.0 genutzt werden
WP_SD_5.3_8	Software-lieferant	SOLL	Es sollen Build-Container („Builder“) für die jeweilige Zielumgebung (Production, Development) verwendet werden. „Builder“ sind temporäre Container, die als Grundlage für den Ziel-Container verwendet werden.

Fortsetzung auf nächster Seite

Fortsetzung von vorheriger Seite

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_5.3_9	Software-lieferant	SOLL	Es soll ein Set an erlaubten Tools für Produktions-Container definiert werden.
WP_SD_5.3_10	Software-lieferant	MUSS	Ein Build muss vollautomatisch erstellt werden, so dass keine Eingriffe in den Build-Prozess erfolgen.
WP_SD_5.3_11	Software-lieferant	SOLL	Es sollen immer die aktuellsten Versionen von Base Images verwendet werden (das sollte durch automatisches Anstoßen der CI/CD-Pipeline erfolgen).
WP_SD_5.3_12	Software-lieferant	MUSS	Ein vorgegebener Satz an Meta-Daten (Label von Images) je Build-Vorgang muss mindestens definiert sein (Versionierung, Ersteller, Repository)
WP_SD_5.3_13	Software-lieferant	SOLL	Für die eindeutige Build-Identifikation soll die Verwendung von Sub-Modulen, Branching oder Tagging erfolgen.
WP_SD_5.3_14	Software-lieferant	SOLL	Es sollen Regelsätze für das Tagging verwendet werden (Informationen zu enthaltenen Komponenten sind in den Metadaten zu speichern)
WP_SD_5.3_15	Software-lieferant	SOLL	Alle (externen) Abhängigkeiten sollten explizit deklariert und isoliert sein (z.B. DB-Image).
WP_SD_5.3_16	Software-lieferant	SOLL	Alle unterstützenden Dienste sollen als Ressource lose gekoppelt werden (z.B. Message Broker).
WP_SD_5.3_17	Software-lieferant	SOLL	Die Build-Erstellung erfolgt vollautomatisch.
DS_10_A003	Software-lieferant	MUSS	Builds werden in separate Phasen unterteilt, mit einer Testphase vor dem Release.
DS_10_A004	Software-lieferant	MUSS	Jeder Build erhält eine eindeutige Kennung und Änderungen werden dokumentiert.
DS_10_A005	Software-lieferant	MUSS	Bei jedem Release wird eine unveränderbare Version erstellt.
DS_10_A006	Software-lieferant	MUSS	Die Build-Erstellung erfolgt vollautomatisch.

3.1.4. Release und Deployment

Für die Veröffentlichung von Releases und deren Installation werden folgende Maßnahmen für Softwarelieferanten definiert:

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_5.4_1	Software-lieferant	SOLL	Es sollen signierte Container (trusted CA) verwendet werden (Sicherstellung, dass das Image von einer vertrauenswürdigen Quelle bezogen wurde).
WP_SD_5.4_2	Software-lieferant	SOLL	Es sollen Build-Regelsätze für das jeweilige Deployment angewendet werden (Best Practices für Build).
WP_SD_5.4_3	Software-lieferant	MUSS	Alle Konfigurationsinformationen für Applikationen müssen außerhalb des Containers gespeichert werden
WP_SD_5.4_4	Software-lieferant	SOLL	Es sollen Kubernetes Config-Maps und Kubernetes Secrets (in verschlüsselter Form) verwendet werden.

Fortsetzung auf nächster Seite

Fortsetzung von vorheriger Seite

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_5.4_5	Software-lieferant	MUSS	Alle Pod-Definitionen müssen Memory- und CPU-Anforderungen enthalten.
WP_SD_5.4_6	Software-lieferant	MUSS	Readiness- und Liveness-Probes müssen in den Definitionen (z.B. bei Deployments oder StatefulSets) enthalten sein.
WP_SD_5.4_7	Software-lieferant	SOLL	Es sollten PodDisruptionBudgets konfiguriert sein, wenn es zwingend erforderlich ist. Single Pod Disruption Budgets sind zu vermeiden.
WP_SD_5.4_8	Software-lieferant	MUSS	Node Affinity Rules sind auf das zwingend erforderliche Maß zu beschränken (z.B. bei Hardware-Abhängigkeiten).
WP_SD_5.4_9	Software-lieferant	SOLL	Alle Pods sollten terminierbar sein, ohne die Stabilität der Applikation zu gefährden. Dabei ist auch auf Session Handling zu achten.
WP_SD_5.4_10	Software-lieferant	SOLL	Die Auto-Scaling Mechanismen sollen – wenn auf der Plattform verfügbar - genutzt werden können.
WP_SD_5.4_11	Software-lieferant	SOLL	Die Deployments sollen auf Basis des Standard-Update (einzeln je Pod), Blue/Green-oder Canary-Modells möglich sein.
WP_SD_5.4_12	Software-lieferant	SOLL	Fehlgeschlagene Deployments sollten über ein Rollback wieder ersetzt werden können.
WP_SD_5.4_13	Software-lieferant	SOLL	Es sollen in einem Container immer nur 1 Dienst und nicht mehrere Dienste (wie z.B. DB und AppServer) laufen.
WP_SD_5.4_14	Software-lieferant	SOLL	Die Application-Logs sollen auf stdout/stderr geschrieben werden, so dass die Zielumgebung für Logs frei gewählt werden kann.
WP_SD_5.4_15	Software-lieferant	SOLL	Es sollen Rate-Limits, Connection-Timeouts und Retries verwendet werden.
WP_SD_5.4_16	Software-lieferant	SOLL	Es soll immer zwischen Build- und Runtime-Images unterschieden werden.
WP_SD_5.4_17	Software-lieferant	MUSS	Pods müssen in einem eingeschränkten Security Context laufen (non-root), Least Privilege Prinzip.
WP_SD_5.4_18	Software-lieferant	SOLL	Für die Kommunikation zwischen Pods soll eine Verschlüsselung gemäß BSI Anforderungen (TR-02102) verwendet werden.
WP_SD_5.4_19	Software-lieferant	SOLL	Es soll ein Zero Trust-Modell verwendet werden und die Kommunikation ausgehend von „deny all“ durch Network-Policies explizit definiert werden.
WP_SD_5.4_20	Software-lieferant	SOLL	Es müssen Rolling-Updates möglich sein.
WP_SD_5.4_21	Software-lieferant	MUSS	Die Updates an Datenstrukturen müssen automatisiert möglich sein. Eine Abwärtskompatibilität der Datenstrukturen (Additive Updates) muss über die laufenden Pods während der Updates sichergestellt sein.
WP_SD_5.4_22	Software-lieferant	SOLL	Updates sollen ohne Betriebsunterbrechung möglich sein. Es bieten sich Microservice-Technologien an.
WP_SD_5.4_23	Software-lieferant	MUSS	Die maximale Anzahl nicht verfügbarer Pods muss definiert werden und darf nicht unterschritten werden.
WP_SD_5.4_24	Software-lieferant	MUSS	Die maximal zulässige Anzahl von Pods und Ressourcen muss so ausgelegt werden, dass Rolling-Updates ausgeführt werden können.

3.1.5. Technische Vorgaben

Zur Vereinheitlichung der Umgebungen werden folgende technische Vorgaben gemacht:

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_5.5_1	Software-lieferant	MUSS	Alle Dienste müssen zustandslos implementiert werden.
WP_SD_5.5_2	Software-lieferant	MUSS	Alle Daten müssen in unterstützenden Diensten gespeichert werden, normalerweise einer Datenbank oder persistent Storage.
WP_SD_5.5_3	Software-lieferant	SOLL	Es sollen keine lokalen Speicher der Workernodes benutzt werden, ausgenommen Ephemeral Storage via EmptyDir Mounts.
WP_SD_5.5_4	Software-lieferant	SOLL	Dateien sollen so gespeichert werden, dass diese ggf. durch andere Container nachgenutzt werden können, wenn der speichernde Container neu gestartet wird.
WP_SD_5.5_5	Software-lieferant	MUSS	Sticky Sessions sind nicht zulässig. Dass Session-Handling muss so implementiert werden, dass die Weitergabe der Sessions zwischen den Pods, z. B. über persistente Speicher, realisiert wird.
WP_SD_5.5_6	Software-lieferant	MUSS	Web- und Applikationsdienste bringen einen eigenen Web- oder Applikationsserver im Container mit.
WP_SD_5.5_7	Software-lieferant	MUSS	Die Kommunikation zwischen den Diensten erfolgt über dokumentierte Ports und werden über Services (Cluster / Namespace intern) oder Ingress Routen (Cluster extern) implementiert.
WP_SD_5.5_8	Software-lieferant	MUSS	Die eingehende Kommunikation im Cluster beginnt am Ingress-Controller.
WP_SD_5.5_9	Software-lieferant	SOLL	Pro Container soll nur ein Dienst laufen.
WP_SD_5.5_10	Software-lieferant	SOLL	Dienste gleicher Art sollen gruppiert werden, damit eine horizontale Skalierung unterstützt wird. Das Skalieren soll auf Basis von Prozessen statt auf Basis von Threads realisiert werden.
WP_SD_5.5_11	Software-lieferant	SOLL	Ein automatisches horizontales Skalieren der Anwendung soll unterstützt werden.
WP_SD_5.5_12	Software-lieferant	SOLL	Health-Checks und Monitoring-Punkte in Containern sollen so implementiert werden, dass die automatischen Steuerungsmechanismen der Containerplattform genutzt werden können.
WP_SD_5.5_13	Software-lieferant	MUSS	Die Grenzen für die Instanziierung der Container müssen implementiert werden, z. B. limits und quotas, cpu, memory, anzahl.
WP_SD_5.5_14	Software-lieferant	SOLL	Die Prozesse sollen mittels asynchroner Eventmodelle kommunizieren, um Deadlocks zu verhindern und die Reaktionen auf Benutzereingaben sicherzustellen.
WP_SD_5.5_15	Software-lieferant	SOLL	Das Verhalten der Skalierung sollte immer überwacht werden, um Anomalien zu ermitteln.
WP_SD_5.5_16	Software-lieferant	MUSS	Zum Schutz der Plattform und anderer Cluster-Nutzer müssen klare Vorgaben für die Ressourcennutzung definiert sein und ein Monitoring- und Alertsystem etabliert werden.
WP_SD_5.5_17	Software-lieferant	MUSS	Die Protokollinformationen werden über stdout und stderr ausgegeben (siehe dazu: Logging Architecture).
WP_SD_5.5_18	Software-lieferant	MUSS	Eine benötigte Revisionssicherheit der Protokollierung wird außerhalb der Containerumgebung sichergestellt.
WP_SD_5.5_19	Software-lieferant	SOLL	Softwarebetreiber und Softwareentwickler sollen auf die Protokollierungslösung im Cluster zugreifen können.

Fortsetzung auf nächster Seite

Fortsetzung von vorheriger Seite

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_5.5_20	Software-lieferant	MUSS	Softwarebetreiber und Softwareentwickler dürfen im regulären Betrieb keine Benutzerinteraktionen über stdout und stderr ausgegeben. Die Protokollierung zur Nachvollziehbarkeit in der Anwendung soll in der Anwendung erfolgen, z. B. in der Datenbank.
WP_SD_5.5_21	Software-lieferant	SOLL	Für die Protokollierung soll der syslog-Standard beziehungsweise standardisierte JSON Formate der Logging Systeme (Splunk, Elastic, Loki) verwendet werden, damit eine einheitliche Auswertbarkeit der Daten erreicht werden kann.
WP_SD_5.5_22	Software-lieferant	SOLL	Debuginformationen sollen im produktiven Betrieb ausschließlich im Rahmen des Problemmanagements zur Fehlersuche zeitlich begrenzt ausgegeben werden. (siehe dazu Logging Architecture)

3.2 Vorgaben für Softwarebetreiber

3.2.1 Release und Deployment

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_6.1_1	Software-betreiber	SOLL	Es sollen signierte Container (trusted CA) verwendet werden (Sicherstellung, dass das Image von einer vertrauenswürdigen Quelle bezogen wurde).
WP_SD_6.1_2	Software-betreiber	SOLL	Es sollen Build-Regelsätze für das jeweilige Deployment angewendet werden (Best Practices für Build).
WP_SD_6.1_3	Software-betreiber	MUSS	Alle Konfigurationsinformationen für Applikationen müssen außerhalb des Containers gespeichert werden.
WP_SD_6.1_4	Software-betreiber	SOLL	Es sollen Kubernetes Config-Maps und Kubernetes Secrets (in verschlüsselter Form) verwendet werden.
WP_SD_6.1_5	Software-betreiber	MUSS	Alle Pod-Definitionen müssen Memory- und CPU-Anforderungen enthalten.
WP_SD_6.1_6	Software-betreiber	MUSS	Readiness- und Liveness-Probes müssen in den Definitionen (z.B. bei Deployments oder StatefulSets) enthalten sein.
WP_SD_6.1_7	Software-betreiber	SOLL	Es sollten PodDisruptionBudgets konfiguriert sein, wenn es zwingend erforderlich ist. Single Pod Disruption Budget sind zu vermeiden.
WP_SD_6.1_8	Software-betreiber	MUSS	Node Affinity Rules sind auf das zwingend erforderliche Maß zu beschränken (z.B. bei Hardware-Abhängigkeiten).
WP_SD_6.1_9	Software-betreiber	SOLL	Alle Pods sollten terminierbar sein, ohne die Stabilität der Applikation zu gefährden. Dabei ist auch auf Session Handling zu achten.
WP_SD_6.1_10	Software-betreiber	SOLL	Die Auto-Scaling Mechanismen sollen – wenn auf der Plattform verfügbar - genutzt werden können.
WP_SD_6.1_11	Software-betreiber	SOLL	Die Deployments sollen auf Basis des Standard-Update (einzeln je Pod), Blue/Green- oder Canary-Modells möglich sein
WP_SD_6.1_12	Software-betreiber	SOLL	Fehlgeschlagene Deployments sollten über ein Rollback wieder ersetzt werden können.
WP_SD_6.1_13	Software-betreiber	SOLL	Es sollen in einem Container immer nur 1 Dienst und nicht mehrere Dienste (wie z.B. DB und AppServer) laufen
WP_SD_6.1_14	Software-betreiber	SOLL	Die Application-Logs sollen auf stdout/stderr geschrieben werden, so dass die Zielumgebung für Logs frei gewählt werden kann.
WP_SD_6.1_15	Software-betreiber	SOLL	Es sollen Rate-Limits, Connection-Timeouts und Retries verwendet werden.
WP_SD_6.1_16	Software-betreiber	SOLL	Es soll zwischen Build- und Runtime-Images unterschieden werden.
WP_SD_6.1_17	Software-betreiber	MUSS	Pods müssen in einen eingeschränkten Security Context laufen (non-root), Least Privilege Prinzip.
WP_SD_6.1_18	Software-betreiber	SOLL	Für die Kommunikation zwischen Pods soll eine Verschlüsselung gemäß BSI-Anforderungen (TR-02102) verwendet werden
WP_SD_6.1_19	Software-betreiber	SOLL	Es soll ein Zero Trust-Modell verwendet werden und die Kommunikation ausgehend von „deny all“ durch Network-Policies explizit definiert werden.
WP_SD_6.1_20	Software-betreiber	MUSS	Es müssen Rolling-Updates möglich sein.

Fortsetzung auf nächster Seite

Fortsetzung von vorheriger Seite

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_6.1_21	Software-betreiber	MUSS	Die Updates an Datenstrukturen müssen automatisiert möglich sein. Eine Abwärtskompatibilität der Datenstrukturen (Additive Updates) muss über die laufenden Pods während der Updates sichergestellt sein.
WP_SD_6.1_22	Software-betreiber	SOLL	Updates sollen ohne Betriebsunterbrechung möglich sein. Es bieten sich Microservice-Technologien an
WP_SD_6.1_23	Software-betreiber	MUSS	Die maximale Anzahl nicht verfügbarer Pods muss definiert werden und darf nicht überschritten werden.
WP_SD_6.1_24	Software-betreiber	MUSS	Die maximal zulässige Anzahl von Pods und Ressourcen muss so ausgelegt werden, dass Rolling-Updates ausgeführt werden können.
WP_SD_6.1_25	Software-betreiber	MUSS	Das Deployment muss vollautomatisch ausgeführt werden können.
WP_SD_6.1_26	Software-betreiber	SOLL	Init Container sollten nur nach Notwendigkeit genutzt werden (z.B. für Initialisierung von Strukturen auf verwendeten Persistent Volumes oder für komplexe Startup Probes)
WP_SD_6.1_27	Software-betreiber	MUSS	Wenn eine Migration beim Start der Anwendung ausgeführt wird, müssen die healthCheck-Mechanismen der Clusterüberwachung ohne Einschränkung unterstützt werden, z.B liveness-Probe.
WP_SD_6.1_28	Software-betreiber	MUSS	Die Nutzung der Konsole (REPL) im Referenz- und Produktivbetrieb ist auf das Notwendigste zu beschränken.
WP_SD_6.1_29	Software-betreiber	MUSS	Einmalig auszuführende Skripte dürfen nicht nachgeladen werden und müssen Bestandteil des Deployments sein.
WP_SD_6.1_30	Software-betreiber	MUSS	Admin-Prozesse laufen gegen ein Release und benutzen dieselbe Codebase und Konfiguration wie jeder Prozess, der gegen ein Release läuft.
WP_SD_6.1_31	Software-betreiber	MUSS	Administrationscode wird mit dem App-Code ausgeliefert, um Synchronisationsprobleme zu vermeiden.
WP_SD_6.1_32	Software-betreiber	SOLL	Der Softwarebetreiber soll die automatische Skalierung der Anwendung umsetzen.
WP_SD_6.1_33	Software-betreiber	MUSS	Softwarelösungen müssen transaktionsorientiert implementiert werden, damit der kurzfristige Ausfall von Containern, z. B. durch Neustart, kompensiert werden kann.
WP_SD_6.1_34	Software-betreiber	SOLL	Sessions sollen beim Neustart der Anwendung erhalten bleiben.
WP_SD_6.1_35	Software-betreiber	SOLL	Es sollen DevSecOps-Prinzipien durchgängig angewendet werden.
WP_SD_6.1_36	Software-betreiber	SOLL	Es soll eine vollständige CI- und CD-Pipeline mit einer nachvollziehbaren Verwaltung der Änderungen aufgebaut und verwendet werden.
WP_SD_6.1_37	Software-betreiber	SOLL	Die Stages Ref, Test, Prod sollen entsprechend den Beschreibungen und den fachlichen und funktionalen Anforderungen aufgebaut sein: - Referenzumgebung - Testumgebung - Produktionsumgebung

Fortsetzung auf nächster Seite

Fortsetzung von vorheriger Seite

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_6.1_38	Software-betreiber	MUSS	Weitere Anforderungen aus dem Maßnahmenkatalog zur Standardisierung zwischen Datenzentralen: - SYS.1.6.A9 Eignung für Container-Betrieb (S) - APP.4.4.A21 Regelmäßiger Restart von Pods (H) - SYS.1.6.A11 Nur ein Dienst pro Container (S) - APP.4.4.A11 Überwachung der Container (S) - SYS.1.6.A12 Verteilung sicherer Images (S) - SYS.1.6.A13 Freigabe von Images (S) - SYS.1.6.A23 Unveränderlichkeit der Container (H) - SYS.1.6.A14 Aktualisierung von Images (S)
WP_SD_6.1_39	Software-betreiber	SOLL	Der Softwarelieferant und der Softwarebetreiber sollen die kritischen Vorgänge identifizieren und eine Auditierfähigkeit der Lösung sicherstellen. Siehe auch Cloud Auditing Data Federation
DS_10_A007	Software-betreiber	MUSS	das Ausbringen sowie Aktualisieren der gelieferten Software planen. Im Fehlerfall ist die Möglichkeit eines Rollbacks vorzuhalten. Um den Betrieb zu verschlanken ist wo möglich eine automatisierte Continous Deployment Pipeline zu erstellen (s. Detailstandard 16: Deployment von Softwareloesungen)

3.2.2 Vorgaben zum Test

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_6.2_1	Software-betreiber	MUSS	Es muss ein Testmanagement für jede Anwendung etabliert werden.
WP_SD_6.2_2	Software-betreiber	SOLL	Die gelieferte Softwareversion soll hinsichtlich der Eignung für den Produktivbetrieb geprüft werden auf die Erfüllung von: - funktionalen Anforderungen und - nicht funktionalen Anforderungen (z. B. Performance, Sicherheit, Architektur).
WP_SD_6.2_3	Software-betreiber	SOLL	Die Tests sollen entlang der Pipeline möglichst automatisiert durchgeführt werden.
WP_SD_6.2_4	Software-betreiber	MUSS	Die Tests müssen fortlaufend bei allen Aktualisierungen im notwendigen, vorher definierten Ausmaß durchgeführt werden.
WP_SD_6.2_5	Software-betreiber	SOLL	Automatisierte Tests sollten so früh wie möglich in der Pipeline angeordnet werden und möglichst vor manuellen Tests durchgeführt werden um Ressourcen zu schonen.
WP_SD_6.2_6	Software-betreiber	MUSS	Am Ende der Tests steht die Freigabe für die Produktion. Dieses kann ggf. auch automatisch erfolgen.
WP_SD_6.2_7	Software-betreiber	MUSS	Es muss klar geregelt und nachvollziehbar sein, wann eine automatisierte Freigabe erfolgen darf, z. B. nur beim Patch von Grundimages.
WP_SD_6.2_8	Software-betreiber	MUSS	Testergebnisse und Freigaben müssen dokumentiert und kommuniziert werden.

3.2.3 Vorgaben für den Betrieb

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_6.3_1	Software-betreiber	MUSS	Es muss ein Monitoring auf Cluster- und auf Namespace-Ebene eingerichtet sein (Admin und User-Monitoring).
WP_SD_6.3_2	Software-betreiber	MUSS	Die applikationsspezifischen Metriken müssen definiert sein. Diese müssen überwacht werden und mittels Schwellwerte Alarmen und Warnungen auslösen
WP_SD_6.3_3	Software-betreiber	MUSS	Es muss ein Logging (STOUT, STDERR) auf Pod-Ebene bzw. auf Cluster-Ebene eingerichtet sein (Admin und User-Logging).
WP_SD_6.3_4	Software-betreiber	SOLL	Das Cluster-Auditing soll per default aktiviert sein und Auffälligkeiten müssen protokolliert werden, so dass entsprechende Alarme verfügbar sind (z. B. für ein SIEM).
WP_SD_6.3_5	Software-betreiber	SOLL	Die Updates des Clusters sollen unterbrechungsfrei durchgeführt werden.
WP_SD_6.3_6	Software-betreiber	SOLL	Es soll sichergestellt werden, dass die Anwendungen bei Cluster-Updates auch unterbrechungsfrei weiterlaufen können.
WP_SD_6.3_7	Software-betreiber	MUSS	Die erforderlichen Security-Updates sind möglichst zeitnahe einzusetzen.
WP_SD_6.3_8	Software-betreiber	SOLL	Applikations-Updates sollen so regelmäßig erfolgen, dass keine deprecated Komponenten verwendet werden müssen.
WP_SD_6.3_9	Software-betreiber	MUSS	Es muss ein Backup-/Recovery-Prozess sowohl auf Cluster- als auch auf Persistenz-Ebene eingesetzt werden.
WP_SD_6.3_10	Software-betreiber	SOLL	Die Datensicherung soll auch die externe Registry und das Git-Repository (Image Copy) umfassen.
WP_SD_6.3_11	Software-betreiber	MUSS	Sämtliche Veränderungen am Cluster und jedes Deployment müssen protokolliert werden.
WP_SD_6.3_12	Software-betreiber	SOLL	Die Protokollierung soll lt. BSI OPS.1.1.5 erfolgen.
WP_SD_6.3_13	Software-betreiber	SOLL	Anwendungen sollen so implementiert werden, dass sie redundant betrieben werden können.
WP_SD_6.3_14	Software-betreiber	SOLL	Anwendungen sollen so implementiert werden, dass Containerinstanzen möglichst klein sind sowie beliebig erzeugt und gelöscht werden können. -> automatische Skalierbarkeit
WP_SD_6.3_15	Software-betreiber	MUSS	Notwendige Ressourcen und Quotas (CPU, RAM, Storage, Anzahl von Pods (Skalierung)) müssen definiert werden. Es muss abgesichert werden, dass ausreichende Ressourcen zur Verfügung stehen und die Quotas nicht überschritten werden können.
WP_SD_6.3_16	Software-betreiber	SOLL	Parameter zur Anomalieerkennung sollten unter Einbeziehung des Dev(Sec)Ops-Teams definiert werden.
WP_SD_6.3_17	Software-betreiber	MUSS	Es muss ein Prozess zum Zertifikatsmanagement inkl. Erstellung, Speicherung und Erneuerung etabliert werden.

3.2.4 Vorgaben zum Logmanagement

Folgende Vorgaben werden im Whitepaper Standardisierung des Deployments für das Logmanagement zur Vereinheitlichung definiert:

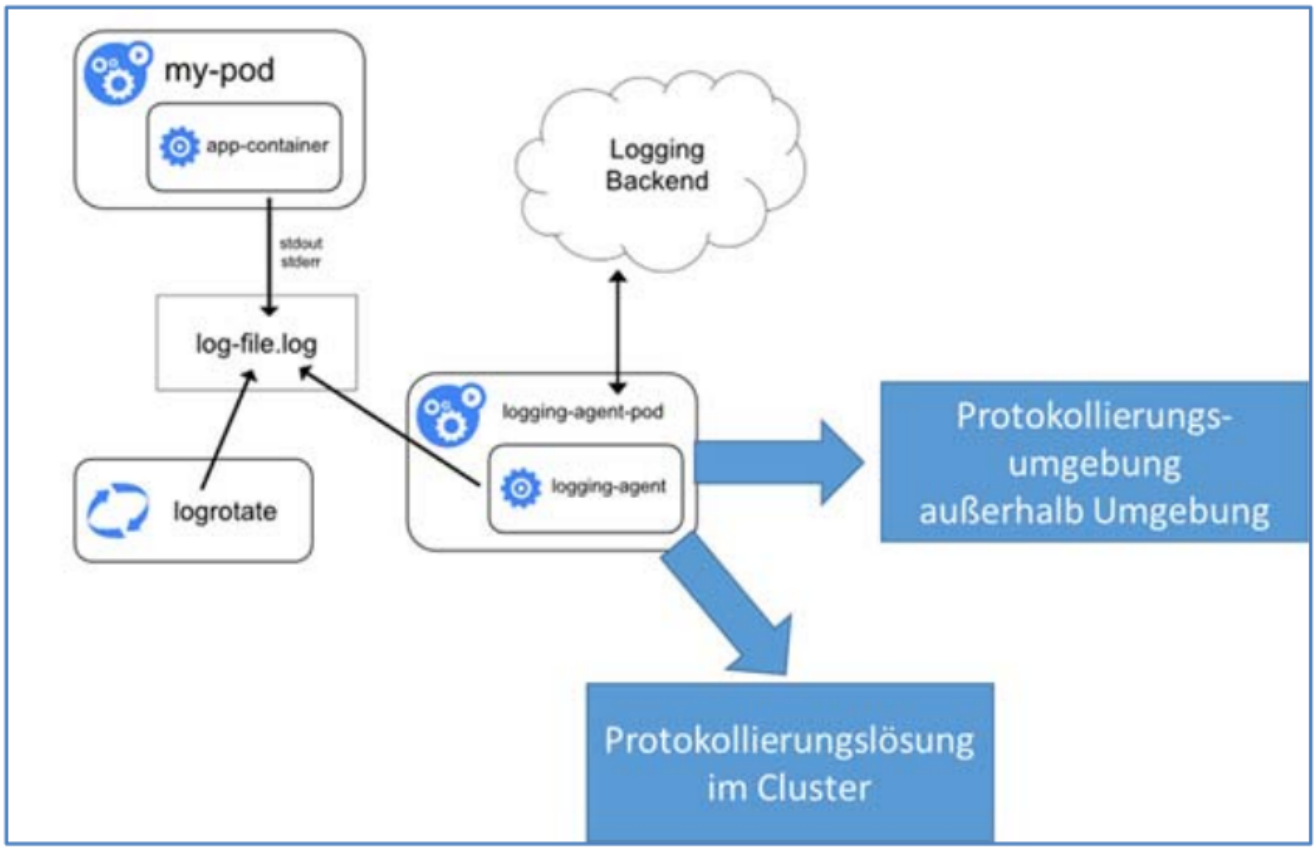

Abbildung 1: Logmanagement

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_6.4_1	Software-betreiber	MUSS	Die Protokollinformationen werden über stdout und stderr ausgegeben. - Dokumentation dazu: Logging Architecture
WP_SD_6.4_2	Software-betreiber	MUSS	Eine benötigte Revisionssicherheit der Protokollierung wird außerhalb der Containerumgebung sichergestellt.
WP_SD_6.4_3	Software-betreiber	SOLL	Softwarebetreiber und –entwickler sollen auf die Protokollierungslösung im Cluster zugreifen können.
WP_SD_6.4_4	Software-betreiber	MUSS	Softwarebetreiber und –entwickler dürfen im regulären Betrieb keine Benutzerinteraktionen über stdout und stderr ausgegeben. Die Protokollierung zur Nachvollziehbarkeit in der Anwendung soll in der Anwendung erfolgen (z.B. in der Datenbank oder über ein applikationsspezifisches Logging).
WP_SD_6.4_5	Software-betreiber	SOLL	Für die Protokollierung soll der syslog-Standard beziehungsweise standardisierte JSON -Formate der Logging Systeme (Splunk, Elastic, Loki) genutzt werden, damit eine einheitliche Auswertbarkeit der Daten erreicht werden kann.

Fortsetzung auf nächster Seite

Fortsetzung von vorheriger Seite

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_6.4_6	Software-betreiber	SOLL	Debuginformationen sollen im produktiven Betrieb ausschließlich im Rahmen des Problemmanagements zur Fehlersuche zeitlich begrenzt ausgegeben werden.

3.2.5 Technische Vorgaben

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_6.5_1	Software-betreiber	MUSS	Alle Dienste müssen zustandslos implementiert werden.
WP_SD_6.5_2	Software-betreiber	MUSS	Alle Daten müssen in unterstützenden Diensten gespeichert werden, normalerweise einer Datenbank oder persistent Storage.
WP_SD_6.5_3	Software-betreiber	SOLL	Es sollen keine lokalen Speicher der Workernodes benutzt werden, ausgenommen Ephemeral Storage via EmptyDir Mounts
WP_SD_6.5_4	Software-betreiber	SOLL	Dateien sollen so gespeichert werden, dass diese ggf. durch andere Container nachgenutzt werden können, wenn der speichernde Container neu gestartet wird.
WP_SD_6.5_5	Software-betreiber	MUSS	Sticky Sessions sind nicht zulässig. Dass Session-Handling muss so implementiert werden, dass die Weitergabe der Sessions zwischen den Pods, z. B. über persistente Speicher, realisiert wird.
WP_SD_6.5_6	Software-betreiber	MUSS	Web- und Applikationsdienste bringen einen eigenen Web- oder Applikationsserver im Container mit.
WP_SD_6.5_7	Software-betreiber	MUSS	Die Kommunikation zwischen den Diensten erfolgt über dokumentierte Ports und werden über Services (Cluster / Namespace intern) oder Ingress Routen (Cluster extern) implementiert.
WP_SD_6.5_8	Software-betreiber	MUSS	Die eingehende Kommunikation im Cluster beginnt am Ingress-Controller.
WP_SD_6.5_9	Software-betreiber	SOLL	Pro Container soll nur ein Dienst laufen.
WP_SD_6.5_10	Software-betreiber	SOLL	Dienste gleicher Art sollen gruppiert werden, damit eine horizontale Skalierung unterstützt wird. Das Skalieren soll auf Basis von Prozessen statt auf Basis von Threads realisiert werden.
WP_SD_6.5_11	Software-betreiber	SOLL	Ein automatisches horizontales Skalieren der Anwendung soll unterstützt werden.
WP_SD_6.5_12	Software-betreiber	SOLL	Health-Checks und Monitoring-Punkte in Containern sollen so implementiert werden, dass die automatischen Steuerungsmechanismen der Containerplattform genutzt werden können.
WP_SD_6.5_13	Software-betreiber	MUSS	Die Grenzen für die Instanzierung der Container müssen implementiert werden, z. B. limits und quotas, cpu, memory, anzahl.
WP_SD_6.5_14	Software-betreiber	SOLL	Die Prozesse sollen mittels asynchroner Eventmodelle kommunizieren, um Deadlocks zu verhindern und die Reaktionen auf Benutzereingaben sicherzustellen.
WP_SD_6.5_15	Software-betreiber	SOLL	Das Verhalten der Skalierung sollte immer überwacht werden, um Anomalien zu ermitteln.

Fortsetzung auf nächster Seite

Fortsetzung von vorheriger Seite

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_6.5_16	Software-betreiber	MUSS	Zum Schutz der Plattform und anderer Cluster-Nutzer müssen klare Vorgaben für die Ressourcennutzung definiert sein und ein Monitoring- und Alert-System etabliert werden.

3.2.6 Messbarkeit

ID	Rolle	Modal-verb	Detailanforderung
WP_SD_6.6_1	Software-betreiber	SOLL	Die für den Betrieb der Container zugewiesenen Rechenressourcen sollten messbar sein, und Unternehmen, die den Cluster nutzen, sollen dafür verantwortlich sein.
WP_SD_6.6_2	Software-betreiber	SOLL	Eine Messfunktion, die die zugewiesenen Rechenressourcen erfasst und diese zur Verrechnung aggregiert und automatisiert bereitstellt, soll verfügbar sein.
WP_SD_6.6_3	Software-betreiber	SOLL	Es sollten wichtige Metriken zum Lebenszyklus der Softwarelösungen gemessen werden (z.B. Releasehäufigkeit, Fehleranfälligkeit etc.).
WP_SD_6.6_4	Software-betreiber	SOLL	Es sollen Service Level Parameter definiert, gemessen und automatisiert berichtet werden.

Anhang

Referenzdokumente

Dokument	Link	PDF
Whitepaper Standardisierung des Deployments	nur PDF	Whitepaper PDF
DVC Detailstandards - (16) Deployment von Softwarelösungen	Detailstandard 16: “Deployment von Softwarelösungen”	Detailstandard 16 PDF