# 15.7 Cloud-Betrieb, Monitoring, Skalierung und Kostenkontrolle

Cloud-Dienste müssen auch nach der Einrichtung betrieben, überwacht und optimiert werden.

Cloud bedeutet nicht:

```
einmal erstellen
und nie wieder anfassen
```

Cloud-Betrieb umfasst:

```
- Monitoring
- Logging
- Alarmierung
- Kostenkontrolle
- Skalierung
- Patchmanagement
- Backup-Prüfung
- Sicherheitsprüfung
- Rechteprüfung
- Dokumentation
- Automatisierung
- Störungsbehandlung
```

Merksatz:

```
Cloud-Ressourcen brauchen laufenden Betrieb wie lokale IT-Systeme.
```

---

**Warum Cloud-Betrieb wichtig ist**

Cloud-Ressourcen lassen sich schnell erstellen.

Das ist ein Vorteil, kann aber auch ein Risiko sein.

Typische Probleme:

Ressourcen werden vergessen.
Kosten steigen unbemerkt.
Logs sind nicht aktiviert.
Backups werden nicht getestet.
Sicherheitsregeln sind zu breit.
Benutzerrechte bleiben zu lange bestehen.
Dienste laufen dauerhaft,
obwohl sie nicht mehr gebraucht werden.

Merksatz:

Cloud-Flexibilität braucht Kontrolle.

---

### Cloud-Betrieb einfach erklärt

Beim Cloud-Betrieb geht es darum, Cloud-Dienste dauerhaft zuverlässig, sicher und wirtschaftlich zu betreiben.

Dazu gehören Fragen wie:

Läuft der Dienst?
Ist der Dienst erreichbar?
Ist die Leistung ausreichend?
Gibt es Sicherheitsereignisse?
Werden Kosten eingehalten?
Funktionieren Backups?
Sind Ressourcen richtig dimensioniert?
Sind Rechte korrekt?
Gibt es unnötige Ressourcen?

Merksatz:

Cloud-Betrieb verbindet Technik,
Sicherheit
und Kosten.

---

### Monitoring

Monitoring bedeutet:

```
Systeme und Dienste werden laufend überwacht.
```

Ziel:

```
Probleme früh erkennen,
bevor Benutzer stark betroffen sind.
```

Überwacht werden zum Beispiel:

```
Verfügbarkeit
CPU-Auslastung
RAM-Auslastung
Speicherplatz
Netzwerkverkehr
Antwortzeiten
Fehlerquoten
Datenbanklast
Zertifikatsablauf
Backupstatus
Kosten
```

Merksatz:

```
Monitoring zeigt,
ob Systeme gesund laufen.
```

---

**Metriken**

Metriken sind messbare Werte.

Beispiele:

```
CPU in Prozent
RAM-Verbrauch
freier Speicherplatz
Anfragen pro Sekunde
```

Antwortzeit in Millisekunden
Fehlerrate
Netzwerkdurchsatz
Anzahl aktiver Benutzer
Datenbankverbindungen
Warteschlangenlänge

Merksatz:

Metriken liefern messbare Zustandswerte.

---

### Logs

Logs sind Ereignisprotokolle.

Sie zeigen, was in einem System passiert ist.

Beispiele:

Benutzeranmeldung
API-Aufruf
Fehler in Anwendung
Firewall blockiert Verbindung
Admin ändert Rolle
Backup schlägt fehl
Datenbank meldet Fehler
Ressource wird gelöscht

Merksatz:

Logs zeigen Ereignisse und helfen bei Fehlersuche.

---

### Monitoring und Logging unterscheiden

| Begriff | Frage | Beispiel |
|---|---|---|
| Monitoring | Wie geht es dem System gerade? | CPU 90 %, Dienst nicht erreichbar |
| Logging | Was ist passiert? | Benutzer hat Firewall-Regel geändert |

Merksatz:

```
Monitoring zeigt Zustand.
Logs zeigen Ereignisse.
```

---

### Alarmierung

Alarmierung bedeutet:

```
Bei bestimmten Ereignissen oder Grenzwerten wird automatisch gewarnt.
```

Beispiele:

```
CPU über 90 %
Speicher fast voll
Dienst nicht erreichbar
Backup fehlgeschlagen
viele fehlgeschlagene Logins
Kostenlimit fast erreicht
Zertifikat läuft bald ab
ungewöhnlich viel Datenverkehr
```

Merksatz:

```
Alarmierung macht Probleme sichtbar,
bevor sie übersehen werden.
```

---

### Schwellwert

Ein Schwellwert ist ein Grenzwert, ab dem eine Aktion ausgelöst wird.

Beispiel:

```
Wenn CPU länger als 10 Minuten über 85 % liegt,
Alarm auslösen.
```

Oder:

```
Wenn Monatskosten 80 % des Budgets erreichen,
Warnung senden.
```

Merksatz:

```
Schwellwerte steuern,
wann gewarnt oder reagiert wird.
```

---

**Alert Fatigue**

Alert Fatigue bedeutet:

```
Es gibt so viele Alarme,
dass wichtige Alarme übersehen werden.
```

Ursachen:

```
zu niedrige Schwellwerte
zu viele unwichtige Alarme
keine Priorisierung
Alarme ohne Handlungsempfehlung
wiederholte Fehlalarme
```

Schutz:

```
Alarme sinnvoll priorisieren
nur relevante Alarme erstellen
klare Zuständigkeiten festlegen
Alarme regelmäßig überprüfen
```

Merksatz:

```
Zu viele schlechte Alarme sind selbst ein Problem.
```

---

**Dashboard**

Ein Dashboard zeigt wichtige Betriebswerte übersichtlich an.

Beispiele:

```
Verfügbarkeit
Auslastung
Fehlerquote
Kosten
offene Alarme
Backupstatus
Zertifikate
Sicherheitsereignisse
aktive Ressourcen
```

Merksatz:

```
Dashboard = Überblick über wichtige Zustände.
```

---

**Cloud-Logging aktivieren**

Viele Cloud-Dienste erzeugen Logs nur dann vollständig, wenn Logging aktiviert oder richtig konfiguriert ist.

Wichtig:

```
Audit-Logs aktivieren
Netzwerklogs aktivieren
Anwendungslogs aktivieren
Firewall-Logs aktivieren
Load-Balancer-Logs aktivieren
Aufbewahrung festlegen
Zugriff auf Logs schützen
```

Merksatz:

```
Nicht vorhandene Logs helfen im Fehlerfall nicht.
```

---

**Audit-Logs**

Audit-Logs dokumentieren administrative Aktionen.

Beispiele:

```
Benutzer erstellt
Rolle geändert
VM gelöscht
Firewall geöffnet
Speicher öffentlich gemacht
API-Schlüssel erzeugt
MFA deaktiviert
Backup gelöscht
```

Merksatz:

```
Audit-Logs zeigen,
wer was wann geändert hat.
```

---

**Flow Logs**

Flow Logs zeigen Netzwerkverbindungen als Metadaten.

Typische Angaben:

```
Quelle
Ziel
Port
Protokoll
Zeit
Aktion
Datenmenge
Schnittstelle
```

Wichtig:

```
Flow Logs zeigen meist nicht den vollständigen Inhalt der Pakete.
```

Merksatz:

```
Flow Logs helfen bei Netzwerk- und Firewall-Analyse.
```

### Log-Aufbewahrung

Logs sollten nicht beliebig kurz oder beliebig lange gespeichert werden.

Zu klären:

```
Wie lange werden Logs benötigt?
Gibt es rechtliche Vorgaben?
Enthalten Logs personenbezogene Daten?
Wer darf Logs lesen?
Wie werden Logs geschützt?
Wann werden Logs gelöscht?
```

Merksatz:

```
Log-Aufbewahrung braucht Sicherheits- und Datenschutzkonzept.
```

---

### Zentrales Logging

Bei mehreren Cloud-Diensten sollten Logs zentral gesammelt werden.

Vorteile:

```
bessere Übersicht
einfachere Suche
Korrelation von Ereignissen
Sicherheitsanalyse
Nachvollziehbarkeit
Alarmierung
weniger Insellösungen
```

Merksatz:

```
Zentrales Logging erleichtert Betrieb und Sicherheitsanalyse.
```

---

### SIEM

SIEM steht für:

```
Security Information and Event Management
```

Ein SIEM sammelt Sicherheitslogs aus verschiedenen Quellen und wertet sie aus.

Ziel:

```
Angriffe erkennen
verdächtige Muster finden
Alarme erzeugen
Ereignisse korrelieren
Nachweise liefern
```

Merksatz:

```
SIEM hilft bei zentraler Sicherheitsüberwachung.
```

---

**Cloud-Kostenkontrolle**

Cloud-Kosten können schnell steigen, wenn Ressourcen unkontrolliert genutzt werden.

Kosten entstehen zum Beispiel durch:

```
laufende virtuelle Maschinen
Speicher
Datenbanken
Backups
Snapshots
Netzwerkverkehr
Load Balancer
öffentliche IP-Adressen
Logs
Monitoring
Lizenzen
Support
Datenübertragung zwischen Regionen
```

Merksatz:

Cloud-Kosten entstehen nicht nur durch Server.

---

### Pay as you go

Pay as you go bedeutet:

bezahlt wird nach Nutzung.

Beispiele:

pro Stunde
pro Sekunde
pro GB
pro Anfrage
pro Benutzer
pro Datenübertragung
pro Ausführung

Vorteil:

flexibel

Risiko:

Fehlkonfiguration oder Dauerbetrieb kann teuer werden.

Merksatz:

Pay as you go braucht Überwachung.

---

### Budget

Ein Budget legt fest, wie viel Geld für Cloud-Ressourcen geplant ist.

Beispiele:

Monatsbudget für Projekt

Budget für Testumgebung

Budget für Abteilung

Budget für gesamte Cloud-Plattform

Wichtig:

Budget allein stoppt nicht immer automatisch Kosten.
Oft braucht man zusätzlich Alarme oder Sperren.

Merksatz:

Budget macht Kosten sichtbar und planbar.

---

**Kostenalarm**

Ein Kostenalarm warnt, wenn Kosten einen bestimmten Wert erreichen.

Beispiele:

50 % des Monatsbudgets erreicht

80 % des Monatsbudgets erreicht

ungewöhnlicher Kostenanstieg

unerwarteter Dienst verursacht Kosten

Merksatz:

Kostenalarme verhindern nicht alles,
aber sie machen Kosten früh sichtbar.

---

**Kostenstelle**

Eine Kostenstelle ordnet Kosten einem Bereich zu.

Beispiele:

```
Projekt
Abteilung
Kunde
Umgebung
Team
Anwendung
```

In der Cloud erfolgt diese Zuordnung oft über:

```
Tags
Labels
Ressourcengruppen
Abonnements
Accounts
Projekte
```

Merksatz:

```
Kosten müssen verursachergerecht zugeordnet werden.
```

---

**Tagging**

Tagging bedeutet:

```
Ressourcen werden mit Metadaten markiert.
```

Beispiele:

```
Projekt = CRM

Umgebung = Produktion

Verantwortlicher = IT-Betrieb

Kostenstelle = 4711

Ablaufdatum = 2026-12-31
```

```
Schutzbedarf = vertraulich
```

Vorteile:

```
Kosten zuordnen
Ressourcen finden
Verantwortliche erkennen
Automatisierung ermöglichen
Aufräumen erleichtern
```

Merksatz:

```
Tags machen Cloud-Ressourcen verwaltbar.
```

---

**Untagged Resources**

Untagged Resources sind Ressourcen ohne Tags.

Problem:

```
Verantwortlicher unklar
Kostenstelle unklar
Zweck unklar
Aufräumen schwierig
Sicherheitsbewertung schwieriger
Lebenszyklus unklar
```

Merksatz:

```
Ressourcen ohne Tags sind im Betrieb schwer kontrollierbar.
```

---

**Rightsizing**

Rightsizing bedeutet:

```
Ressourcen passend dimensionieren.
```

Beispiel:

```
Eine VM hat 16 vCPU und 64 GB RAM,
nutzt aber dauerhaft nur 5 %.
```

Dann ist sie wahrscheinlich zu groß.

Maßnahmen:

```
kleinere Instanz wählen
Datenbankklasse anpassen
Speicherklasse ändern
ungenutzte Ressourcen löschen
Lastverhalten prüfen
```

Merksatz:

```
Rightsizing senkt Kosten durch passende Größe.
```

---

**Overprovisioning**

Overprovisioning bedeutet:

```
Ressourcen sind größer bereitgestellt,
als tatsächlich benötigt.
```

Beispiele:

```
zu große VM
zu teure Datenbankklasse
zu viel reservierter Speicher
zu viele laufende Instanzen
unnötig hohe Performanceklasse
```

Merksatz:

```
Overprovisioning kostet unnötig Geld.
```

---

### Underprovisioning

Underprovisioning bedeutet:

```
Ressourcen sind zu klein dimensioniert.
```

Folgen:

```
langsame Anwendung
Ausfälle
hohe Antwortzeiten
Datenbankengpässe
Fehler bei Lastspitzen
schlechte Benutzererfahrung
```

Merksatz:

```
Zu klein spart kurzfristig Geld,
kann aber Betrieb stören.
```

---

### Auto Scaling

Auto Scaling bedeutet:

```
Ressourcen werden automatisch erhöht oder reduziert.
```

Beispiel:

```
Bei hoher Last werden zusätzliche Instanzen gestartet.

Bei geringer Last werden Instanzen beendet.
```

Vorteile:

```
bessere Anpassung an Last
weniger manuelle Eingriffe
bessere Verfügbarkeit
```

Kosten können optimiert werden

Merksatz:

Auto Scaling passt Ressourcen automatisch an Last an.

---

### Horizontale Skalierung

Horizontale Skalierung bedeutet:

Es werden mehr Instanzen hinzugefügt.

Beispiel:

statt 1 Webserver
laufen 3 Webserver

Vorteil:

Last kann verteilt werden
Ausfallsicherheit steigt
gut mit Load Balancer kombinierbar

Merksatz:

Horizontal skalieren = mehr Systeme.

---

### Vertikale Skalierung

Vertikale Skalierung bedeutet:

Ein einzelnes System wird größer gemacht.

Beispiel:

VM bekommt mehr CPU
VM bekommt mehr RAM
Datenbank bekommt größere Leistungsklasse

Vorteil:

oft einfacher

Nachteil:

Grenze durch maximale Größe
eventuell Neustart nötig
einzelnes System bleibt kritischer

Merksatz:

Vertikal skalieren = stärkeres System.

---

### Horizontale und vertikale Skalierung vergleichen

| Skalierungsart | Bedeutung | Beispiel |
|---|---|---|
| horizontal | mehr Instanzen | drei Webserver statt einer |
| vertikal | größere Instanz | mehr CPU und RAM für eine VM |

Merksatz:

Horizontal = mehr.
Vertikal = stärker.

---

### Scale Out und Scale Up

Scale Out bedeutet:

horizontal skalieren

Scale Up bedeutet:

vertikal skalieren

Beispiele:

Scale Out:
mehr App-Instanzen starten

Scale Up:
größere Datenbankklasse wählen

Merksatz:

Scale Out = mehr Instanzen.
Scale Up = größere Instanz.

---

**Scale In und Scale Down**

Scale In bedeutet:

Instanzen reduzieren

Scale Down bedeutet:

einzelne Instanz verkleinern

Beispiele:

Scale In:
von 5 Webservern auf 2 Webserver reduzieren

Scale Down:
VM von großer auf kleinere Größe ändern

Merksatz:

```
Scale In reduziert Anzahl.
Scale Down reduziert Größe.
```

---

### Skalierungsregel

Eine Skalierungsregel legt fest, wann skaliert wird.

Beispiele:

```
Wenn CPU länger als 10 Minuten über 70 %,
dann eine Instanz hinzufügen.

Wenn Warteschlange mehr als 100 Nachrichten hat,
dann Worker erhöhen.

Wenn CPU länger niedrig,
dann Instanz entfernen.
```

Merksatz:

```
Skalierung braucht sinnvolle Regeln.
```

---

### Skalierung ist nicht nur CPU

Skalierung kann abhängig sein von:

```
CPU
RAM
Anzahl Anfragen
Antwortzeit
Warteschlangenlänge
Datenbankverbindungen
Netzwerkdurchsatz
Fehlerrate
Benutzerzahl
Zeitplan
```

Merksatz:

```
CPU ist nur eine mögliche Skalierungsmetrik.
```

---

### Load Balancer und Skalierung

Bei horizontaler Skalierung verteilt ein Load Balancer den Verkehr.

Ablauf:

```
neue Instanz startet
Health Check prüft Instanz
Load Balancer nimmt Instanz auf
Verkehr wird verteilt
```

Wenn Instanz ungesund ist, wird sie nicht genutzt.

Merksatz:

```
Auto Scaling und Load Balancer arbeiten oft zusammen.
```

---

### Health Check im Betrieb

Ein Health Check prüft, ob ein Dienst gesund ist.

Beispiele:

```
HTTP 200 auf /health
TCP-Port erreichbar
Datenbankverbindung möglich
Anwendung meldet Status „ok“
```

Merksatz:

```
Health Check sollte echte Dienstfähigkeit prüfen,
nicht nur offenen Port.
```

---

### Hochverfügbarkeit

Hochverfügbarkeit bedeutet:

Ein Dienst bleibt möglichst verfügbar,
auch wenn einzelne Komponenten ausfallen.

Mittel:

mehrere Instanzen
mehrere Zonen
Load Balancer
Replikation
automatischer Neustart
Monitoring
Backup
Notfallplan

Merksatz:

Hochverfügbarkeit muss geplant und getestet werden.

---

**Verfügbarkeitszone**

Eine Verfügbarkeitszone ist ein getrenntes Rechenzentrumssegment innerhalb einer Region.

Vorteil:

Fällt eine Zone aus,
können andere Zonen weiterarbeiten.

Cloud-Dienste sollten für hohe Verfügbarkeit oft über mehrere Zonen verteilt werden.

Merksatz:

Mehrere Zonen erhöhen Ausfallsicherheit innerhalb einer Region.

---

**Region**

Eine Region ist ein geografischer Standort eines Cloud-Anbieters.

Regionen beeinflussen:

```
Latenz
Datenstandort
Verfügbarkeit
Kosten
Datenschutz
Compliance
```

Merksatz:

```
Region bewusst nach Technik,
Recht
und Kosten auswählen.
```

---

### Disaster Recovery

Disaster Recovery bedeutet:

```
Wiederherstellung nach schwerem Ausfall.
```

Beispiele:

```
Region fällt aus
Datenbank beschädigt
Ransomware-Angriff
Fehlkonfiguration löscht Ressourcen
kritischer Dienst fällt aus
```

Dazu braucht man:

```
Backups
Wiederherstellungsplan
Ersatzumgebung
Dokumentation
Tests
klare Zuständigkeiten
```

Merksatz:

```
Disaster Recovery muss vor dem Notfall geplant werden.
```

---

### RTO im Cloud-Betrieb

RTO steht für:

```
Recovery Time Objective
```

Es beschreibt, wie lange ein Dienst maximal ausfallen darf.

Beispiel:

```
RTO = 2 Stunden
```

Dann soll der Dienst innerhalb von 2 Stunden wieder verfügbar sein.

Merksatz:

```
RTO beschreibt erlaubte Wiederherstellungszeit.
```

---

### RPO im Cloud-Betrieb

RPO steht für:

```
Recovery Point Objective
```

Es beschreibt, wie viel Datenverlust maximal akzeptiert wird.

Beispiel:

```
RPO = 15 Minuten
```

Dann dürfen höchstens Daten der letzten 15 Minuten verloren gehen.

Merksatz:

```
RPO beschreibt erlaubten Datenverlust.
```

---

### RTO und RPO im Betrieb vergleichen

| Begriff | Frage |
| --- | --- |
| RTO | Wie lange darf der Dienst ausfallen? |
| RPO | Wie viele Daten dürfen verloren gehen? |

Merksatz:

```
RTO = Zeit bis Wiederherstellung.
RPO = Datenverlust bis Wiederherstellungspunkt.
```

---

### Patchmanagement in der Cloud

Auch Cloud-Systeme brauchen Updates.

Wer verantwortlich ist, hängt vom Service-Modell ab.

IaaS:

```
Kunde patcht Betriebssystem und Anwendung.
```

PaaS:

```
Anbieter patcht Plattform,
Kunde patcht Anwendung und Abhängigkeiten.
```

SaaS:

```
Anbieter patcht Anwendung,
Kunde prüft Einstellungen,
Benutzer
und Sicherheit.
```

Merksatz:

```
Patchverantwortung hängt vom Cloud-Modell ab.
```

---

### Wartungsfenster

Ein Wartungsfenster ist ein geplanter Zeitraum für Änderungen oder Updates.

Ziel:

```
Änderungen kontrolliert durchführen
Benutzer informieren
Ausfallzeit planen
Rollback vorbereiten
Risiken reduzieren
```

Merksatz:

```
Wartungsfenster schaffen planbare Änderungen.
```

---

**Change Management**

Change Management bedeutet:

```
Änderungen werden geplant,
geprüft,
freigegeben,
umgesetzt
und dokumentiert.
```

Cloud-Beispiele:

```
neue Firewall-Regel
größere Datenbankklasse
neue Region
neue Rolle
neue Backup-Regel
neue Skalierungsregel
neues Deployment
```

Merksatz:

```
Cloud-Änderungen brauchen Kontrolle und Dokumentation.
```

---

### Rollback

Rollback bedeutet:

```
Eine Änderung wird zurückgenommen.
```

Beispiele:

```
vorherige Anwendungsversion wiederherstellen
alte Konfiguration laden
Snapshot zurücksetzen
Firewall-Regel zurücknehmen
Deployment rückgängig machen
```

Merksatz:

```
Jede kritische Änderung braucht einen Rückweg.
```

---

### Automatisierung im Cloud-Betrieb

Cloud-Betrieb wird häufig automatisiert.

Beispiele:

```
Ressourcen automatisch bereitstellen
Backups automatisch erstellen
Skalierung automatisch durchführen
Alarme automatisch auslösen
Tests automatisch starten
Deployments automatisch ausrollen
ungenutzte Ressourcen automatisch stoppen
```

Merksatz:

```
Automatisierung reduziert manuelle Fehler,
braucht aber gute Regeln.
```

---

### Infrastructure as Code

Infrastructure as Code wird oft abgekürzt:

```
IaC
```

Dabei wird Infrastruktur nicht manuell im Portal zusammengeklickt, sondern als Code beschrieben.

Beispiele:

```
Netzwerke
Subnetze
Firewall-Regeln
virtuelle Maschinen
Datenbanken
Rollen
Speicher
Load Balancer
```

Vorteile:

```
wiederholbar
dokumentiert
versionierbar
prüfbar
automatisierbar
```

Merksatz:

```
IaC beschreibt Infrastruktur als Code.
```

---

**Configuration Drift**

Configuration Drift bedeutet:

```
Die tatsächliche Konfiguration weicht von der geplanten Konfiguration ab.
```

Ursachen:

manuelle Änderungen
Notfalländerungen
unterschiedliche Umgebungen
unvollständige Dokumentation
fehlende Automatisierung

Problem:

Fehlersuche wird schwieriger.
Sicherheit kann unbemerkt schlechter werden.

Merksatz:

Configuration Drift entsteht durch ungeplante Abweichungen.

---

**CI/CD im Cloud-Betrieb**

CI/CD steht für:

Continuous Integration
Continuous Delivery
oder
Continuous Deployment

Ziel:

Änderungen werden automatisiert gebaut,
getestet
und bereitgestellt.

Cloud-Bezug:

Anwendungen werden schneller und kontrollierter ausgeliefert.
Rollbacks können einfacher werden.
Tests werden automatisiert.
Infrastruktur kann mitbereitgestellt werden.

Merksatz:

```
CI/CD automatisiert Entwicklung,
Test
und Bereitstellung.
```

---

### Blue-Green Deployment

Blue-Green Deployment bedeutet:

```
Es gibt zwei Umgebungen.

Blue:
aktuelle produktive Version

Green:
neue Version
```

Nach erfolgreichem Test wird der Verkehr auf die neue Umgebung umgeschaltet.

Vorteil:

```
schneller Rollback möglich
```

Merksatz:

```
Blue-Green reduziert Risiko bei neuen Versionen.
```

---

### Canary Deployment

Canary Deployment bedeutet:

```
Eine neue Version wird zuerst nur für einen kleinen Teil der Benutzer freigegeben.
```

Wenn keine Probleme auftreten, wird der Anteil erhöht.

Vorteil:

```
Fehler betreffen zunächst nur wenige Benutzer.
```

Merksatz:

```
Canary testet neue Versionen schrittweise im Echtbetrieb.
```

---

**Cloud-Betriebsdokumentation**

Dokumentiert werden sollten:

```
Zweck der Ressource
Verantwortlicher
Kostenstelle
Umgebung
Datenklasse
Netzverbindungen
Backup-Regeln
Monitoring
Alarme
Wiederherstellungsplan
Sicherheitsregeln
Änderungen
Abhängigkeiten
```

Merksatz:

```
Was nicht dokumentiert ist,
ist im Betrieb schwer beherrschbar.
```

---

**Runbook**

Ein Runbook ist eine Schritt-für-Schritt-Anleitung für wiederkehrende Betriebsaufgaben.

Beispiele:

```
Dienst neu starten
Backup wiederherstellen
```

Zertifikat erneuern
Alarm bearbeiten
Instanz skalieren
Failover auslösen
Benutzerzugriff prüfen

Merksatz:

Runbooks helfen,
im Betrieb einheitlich zu handeln.

---

**Incident Response**

Incident Response bedeutet:

Vorgehen bei Sicherheitsvorfällen oder größeren Störungen.

Schritte:

erkennen
bewerten
eindämmen
beseitigen
wiederherstellen
dokumentieren
verbessern

Cloud-Beispiele:

kompromittierter API-Schlüssel
öffentlich freigegebener Speicher
ungewöhnliche Kosten
Adminzugang missbraucht
Datenabfluss
Ransomware

Merksatz:

```
Incident Response muss vorbereitet sein,
bevor ein Vorfall passiert.
```

---

### Service Level Agreement

Service Level Agreement wird abgekürzt:

```
SLA
```

Ein SLA beschreibt zugesicherte Servicequalität.

Beispiele:

```
Verfügbarkeit
Reaktionszeit
Supportzeit
Wiederherstellungszeit
Leistungswerte
```

Merksatz:

```
SLA beschreibt,
welche Leistung zugesichert wird.
```

---

### SLO

SLO steht für:

```
Service Level Objective
```

Ein SLO ist ein internes oder technisches Ziel für einen Dienst.

Beispiel:

```
99,9 % erfolgreiche Anfragen pro Monat

Antwortzeit unter 300 ms für 95 % der Anfragen
```

Merksatz:

```
SLO beschreibt ein Ziel für Servicequalität.
```

---

### SLI

SLI steht für:

```
Service Level Indicator
```

Ein SLI ist ein Messwert, mit dem ein SLO bewertet wird.

Beispiele:

```
Verfügbarkeit
Fehlerrate
Antwortzeit
Durchsatz
Latenz
```

Merksatz:

```
SLI ist der Messwert,
SLO ist das Ziel.
```

---

### SLA, SLO und SLI vergleichen

| Begriff | Bedeutung |
|---|---|
| SLA | vertragliche Zusage |
| SLO | Zielwert für Dienstqualität |
| SLI | Messwert zur Bewertung |

Merksatz:

```
SLI misst.
SLO setzt Ziel.
SLA ist Zusage.
```

### Cloud-Fehlersuche im Betrieb

Bei Cloud-Störungen systematisch prüfen:

```
Was ist betroffen?
Seit wann?
Wurde etwas geändert?
Welche Region?
Welche Zone?
Welche Ressource?
Welche Logs?
Welche Metriken?
Welche Abhängigkeiten?
Welche Kostenänderung?
Welche Alarme?
Welche Benutzer betroffen?
```

Merksatz:

```
Störungen immer mit Zeit,
Änderung
Metriken
Logs
und Abhängigkeiten prüfen.
```

---

### Typische Betriebsprobleme in der Cloud

Häufige Probleme:

```
Dienst nicht erreichbar
Zertifikat abgelaufen
Speicher voll
Datenbank überlastet
Kosten steigen plötzlich
Backup fehlgeschlagen
Load Balancer hat keine gesunden Ziele
Security Group falsch geändert
API-Schlüssel kompromittiert
```

Rechte zu breit vergeben
Region oder Zone gestört
Skalierung greift nicht

Merksatz:

Cloud-Betrieb betrifft Verfügbarkeit,
Sicherheit
Leistung
und Kosten.

---

### Checkliste: Cloud-Betrieb

Monitoring aktivieren.
Logs aktivieren.
Alarme einrichten.
Kostenalarme setzen.
Tags pflegen.
Verantwortliche dokumentieren.
Backups prüfen.
Restore-Tests durchführen.
Rechte regelmäßig prüfen.
Ressourcen richtig dimensionieren.
ungenutzte Ressourcen löschen.
Patchverantwortung klären.
Runbooks erstellen.
Änderungen dokumentieren.
Incident Response vorbereiten.

Merksatz:

Cloud-Betrieb braucht regelmäßige Kontrolle.

---

### Typische IHK-Fragen

In AP1 und AP2 kann zum Beispiel gefragt werden:

- Warum braucht Cloud Monitoring?
- Was ist der Unterschied zwischen Monitoring und Logging?
- Was ist ein Kostenalarm?
- Warum ist Tagging sinnvoll?
- Was bedeutet Rightsizing?
- Was ist Overprovisioning?
- Was ist Auto Scaling?
- Was ist horizontale Skalierung?
- Was ist vertikale Skalierung?
- Was ist ein Health Check?
- Was ist der Unterschied zwischen RTO und RPO?
- Was ist Infrastructure as Code?
- Was bedeutet Configuration Drift?
- Was ist ein Runbook?
- Was bedeutet Incident Response?
- Was ist der Unterschied zwischen SLA, SLO und SLI?
- Warum ist Cloud nicht automatisch wartungsfrei?

---

### Typische Prüfungsfallen

Cloud ist nicht wartungsfrei.

Monitoring zeigt Zustand.

Logs zeigen Ereignisse.

Ohne Logs keine gute Fehlersuche.

Zu viele Alarme können schaden.

Kosten entstehen auch durch Speicher,
Traffic,
Logs
und Snapshots.

Pay as you go braucht Kontrolle.

Tags helfen bei Kosten und Verantwortung.

Untagged Resources sind problematisch.

Rightsizing spart Kosten.

Overprovisioning kostet unnötig Geld.

Underprovisioning stört Betrieb.

Auto Scaling braucht sinnvolle Regeln.

Horizontal bedeutet mehr Instanzen.

Vertikal bedeutet größere Instanz.

Hochverfügbarkeit muss geplant werden.

Cloud ist nicht automatisch hochverfügbar.

RTO betrifft Wiederherstellungszeit.

RPO betrifft Datenverlust.

Patchverantwortung hängt vom Service-Modell ab.

Change Management gilt auch in der Cloud.

Rollback vor Änderung planen.

IaC macht Infrastruktur wiederholbar.

Configuration Drift vermeiden.

Runbooks helfen im Betrieb.

SLA,
SLO
und SLI nicht verwechseln.

## Wichtige Begriffe kurz erklärt

| Begriff | Kurze Erklärung |
|---|---|
| Cloud-Betrieb | laufende Verwaltung von Cloud-Diensten |
| Monitoring | Überwachung von Zuständen |
| Metrik | messbarer Wert |
| Logging | Protokollierung von Ereignissen |
| Audit-Log | Protokoll administrativer Aktionen |
| Flow Log | Netzwerkverbindungsdaten |
| Alarmierung | automatische Warnung |
| Schwellwert | Grenzwert für Alarm oder Aktion |
| Alert Fatigue | Überforderung durch zu viele Alarme |
| Dashboard | Übersicht wichtiger Werte |
| SIEM | zentrale Sicherheitslog-Auswertung |
| Kostenkontrolle | Überwachung und Steuerung von Kosten |
| Pay as you go | Abrechnung nach Nutzung |
| Budget | geplanter Kostenrahmen |
| Kostenalarm | Warnung bei Kostenüberschreitung |
| Kostenstelle | Zuordnung von Kosten |
| Tagging | Metadaten an Ressourcen |
| Rightsizing | passende Ressourcengröße |
| Overprovisioning | zu große Ressourcen |
| Underprovisioning | zu kleine Ressourcen |
| Auto Scaling | automatische Anpassung von Ressourcen |
| horizontale Skalierung | mehr Instanzen |
| vertikale Skalierung | größere Instanz |
| Scale Out | horizontal erweitern |
| Scale Up | vertikal vergrößern |
| Health Check | Prüfung der Dienstgesundheit |
| Hochverfügbarkeit | hohe Verfügbarkeit trotz Fehler |
| Disaster Recovery | Wiederherstellung nach schwerem Ausfall |
| RTO | maximale Wiederherstellungszeit |
| RPO | maximaler Datenverlust |

| Begriff | Kurze Erklärung |
|---|---|
| Patchmanagement | Verwaltung von Updates |
| Change Management | kontrollierte Änderungen |
| Rollback | Änderung zurücknehmen |
| IaC | Infrastructure as Code |
| Configuration Drift | Abweichung von Soll-Konfiguration |
| CI/CD | automatisierte Bereitstellung |
| Runbook | Betriebsanleitung |
| Incident Response | Reaktion auf Vorfälle |
| SLA | vertragliche Servicezusage |
| SLO | Zielwert für Servicequalität |
| SLI | Messwert für Servicequalität |

---

### IHK-sichere Kurzformulierung

Cloud-Betrieb umfasst die laufende Überwachung, Protokollierung, Alarmierung, Kostenkontrolle, Skalierung, Wartung, Sicherheitsprüfung und Dokumentation von Cloud-Diensten. Monitoring zeigt den aktuellen Zustand eines Systems, während Logging Ereignisse und Änderungen nachvollziehbar macht. Kostenkontrolle ist wichtig, weil Cloud-Ressourcen nach Nutzung abgerechnet werden und auch Speicher, Netzwerkverkehr, Backups, Logs, Snapshots und Lizenzen Kosten verursachen können. Auto Scaling passt Ressourcen automatisch an Last an. Horizontale Skalierung bedeutet mehr Instanzen, vertikale Skalierung bedeutet größere Instanzen. Für Ausfallsicherheit sind Hochverfügbarkeit, Backups, RTO, RPO, Runbooks, Change Management und Incident Response wichtig.

---

### Merksätze

Cloud ist nicht wartungsfrei.

Cloud-Betrieb braucht Monitoring.

Monitoring zeigt Zustand.

Logs zeigen Ereignisse.

Audit-Logs zeigen Adminaktionen.

Flow Logs zeigen Netzwerkverbindungen.

Alarmierung macht Probleme sichtbar.

Zu viele Alarme können wichtige Alarme verdecken.

Dashboard gibt Überblick.

SIEM wertet Sicherheitslogs zentral aus.

Cloud-Kosten aktiv überwachen.

Pay as you go braucht Kontrolle.

Kosten entstehen auch durch Traffic,
Speicher,
Logs
und Snapshots.

Budget planen.

Kostenalarme setzen.

Tags pflegen.

Untagged Resources vermeiden.

Rightsizing spart Kosten.

Overprovisioning kostet unnötig.

Underprovisioning gefährdet Betrieb.

Auto Scaling passt Ressourcen an.

Horizontal = mehr Instanzen.

Vertikal = stärkere Instanz.

Scale Out = mehr.

Scale Up = größer.

Load Balancer verteilt Last.

Health Check prüft Dienstgesundheit.

Hochverfügbarkeit planen.

Cloud ist nicht automatisch hochverfügbar.

Region und Zone bewusst wählen.

Disaster Recovery vorher planen.

RTO = Wiederherstellungszeit.

RPO = Datenverlust.

Patchverantwortung hängt vom Service-Modell ab.

Änderungen dokumentieren.

Rollback vor Änderung planen.

IaC beschreibt Infrastruktur als Code.

Configuration Drift vermeiden.

CI/CD automatisiert Bereitstellung.

Runbooks helfen im Betrieb.

Incident Response vorbereiten.

SLI misst.

SLO setzt Ziel.

SLA ist Zusage.