13.6 Failover, Quorum und Split-Brain

Einordnung

Diese Seite gehört zum Kapitel:

Server, Virtualisierung und Hochverfügbarkeit

Auf der vorherigen Seite ging es allgemein um Cluster und Hochverfügbarkeit.

Jetzt geht es genauer um drei besonders wichtige Cluster-Begriffe:

Begriff	Bedeutung
Failover	Übernahme bei Ausfall
Quorum	Entscheidungsmechanismus im Cluster
Split-Brain	Gefährlicher Zustand durch getrennte Cluster-Teile

Diese Begriffe sind wichtig, weil sie erklären, wie ein Cluster bei Fehlern entscheidet und warum falsche Entscheidungen zu Datenverlust führen können.

Grundidee

Ein Cluster besteht aus mehreren Servern.

Diese Server müssen ständig wissen:

Frage	Warum wichtig?
Welche Nodes sind erreichbar?	Damit Ausfälle erkannt werden
Welche Ressourcen laufen wo?	Damit Dienste nicht doppelt starten
Wer darf aktiv sein?	Damit keine Daten beschädigt werden
Gibt es eine Mehrheit?	Damit der Cluster eindeutig entscheiden kann

Wenn ein Node ausfällt, soll der Cluster reagieren.

Wenn aber nur die Netzwerkverbindung gestört ist, darf der Cluster nicht einfach falsch übernehmen.

Genau dafür sind Quorum und Split-Brain-Schutz wichtig.

Was ist Failover?

Failover bedeutet, dass ein Dienst oder eine Ressource bei Ausfall eines Systems auf ein anderes System wechselt oder dort neu gestartet wird.

Beispiel:

Eine virtuelle Maschine läuft auf Host 1.

Host 1 fällt aus.

Der Cluster startet die VM auf Host 2 neu.

Schritt	Ereignis
1	VM läuft auf HV01
2	HV01 fällt aus
3	Cluster erkennt den Ausfall
4	VM wird auf HV02 gestartet
5	Dienst ist wieder erreichbar

Merksatz

Failover bedeutet Übernahme bei Ausfall.

Warum Failover wichtig ist

Ohne Failover müsste ein Administrator einen Dienst manuell auf einem anderen System starten.

Das dauert länger und erhöht die Ausfallzeit.

Mit Failover kann ein Cluster schneller reagieren.

Vorteile:

Vorteil	Erklärung
Kürzere Ausfallzeit	Dienst wird schneller wieder verfügbar
Automatische Reaktion	Kein sofortiges manuelles Eingreifen nötig
Bessere Verfügbarkeit	Kritische Dienste fallen weniger lange aus
Geplanter Betrieb	Ressourcen können gezielt verteilt werden

Failover bei virtuellen Maschinen

In einem Virtualisierungscluster kann eine VM bei Host-Ausfall auf einem anderen Host neu gestartet werden.

Beispiel:

VM	Vor Ausfall	Nach Failover
DC01	HV01	HV02
FILE01	HV01	HV03
WEB01	HV02	HV02
DB01	HV03	HV03

Wenn HV01 ausfällt, werden die betroffenen VMs auf anderen Hosts neu gestartet.

Wichtig:

Es gibt meistens eine kurze Unterbrechung, weil die VM neu starten muss.

Failover ist nicht immer unterbrechungsfrei

Failover bedeutet nicht automatisch, dass Benutzer gar nichts merken.

Beispiel:

Situation	Auswirkung
Host fällt aus	VM läuft nicht mehr
Cluster erkennt Ausfall	Fehler wird verarbeitet
VM wird auf anderem Host gestartet	Bootvorgang dauert
Dienst startet	Benutzer können wieder zugreifen

Je nach Dienst kann die Unterbrechung Sekunden bis Minuten dauern.

Failover vs. Fault Tolerance

Begriff	Bedeutung
Failover	Dienst wird nach Ausfall auf anderem System gestartet
Fault Tolerance	Dienst läuft trotz Ausfall möglichst ohne Unterbrechung weiter

Failover ist in vielen IT-Umgebungen üblich.

Fault Tolerance ist technisch aufwendiger.

Automatisches Failover

Beim automatischen Failover erkennt der Cluster einen Fehler selbst und startet die betroffene Ressource auf einem anderen Node.

Beispiel:

Ereignis	Reaktion
Node fällt aus	Cluster startet Ressource auf anderem Node
VM reagiert nicht	Cluster versucht Neustart
Dienst ist nicht erreichbar	Cluster löst Failover aus

Vorteil:

Schnelle Reaktion.

Nachteil:

Wenn die Fehlererkennung falsch ist, kann ein unnötiges Failover ausgelöst werden.

Manuelles Failover

Beim manuellen Failover verschiebt ein Administrator eine Ressource bewusst auf einen anderen Node.

Typische Gründe:

Grund	Beispiel
Wartung	Ein Host soll aktualisiert werden
Test	Failover-Funktion prüfen
Lastverteilung	Ressource soll auf anderen Node
Fehleranalyse	Dienst soll kontrolliert verschoben werden

Manuelles Failover ist kontrollierter als automatisches Failover.

Geplantes Failover

Ein geplantes Failover findet bewusst statt.

Beispiel:

Ein Administrator möchte HV01 warten.

Vorher werden die VMs kontrolliert auf HV02 und HV03 verschoben.

Schritt	Erklärung
1	Wartung ankündigen
2	Backups prüfen
3	VMs verschieben
4	Host warten
5	Funktion prüfen
6	Host wieder aktivieren

Geplantes Failover verursacht meist weniger Risiko als ungeplantes Failover.

Ungeplantes Failover

Ein ungeplantes Failover passiert durch einen Fehler.

Beispiele:

Ursache	Beispiel
Hardwaredefekt	Host fällt plötzlich aus
Stromausfall	Node geht aus
Netzwerkfehler	Node ist nicht erreichbar
Hypervisor-Absturz	Host reagiert nicht
Storageproblem	VM-Daten nicht erreichbar

Ungeplantes Failover ist riskanter, weil der Ausfall plötzlich passiert.

Failback

Failback bedeutet, dass eine Ressource nach der Reparatur wieder auf das ursprüngliche System zurückwechselt.

Beispiel:

Schritt	Ereignis
1	VM läuft auf HV01
2	HV01 fällt aus
3	VM wird auf HV02 gestartet

Schritt	Ereignis
4	HV01 wird repariert
5	VM wird wieder zurück auf HV01 verschoben

Failback kann automatisch oder manuell erfolgen.

Automatisches Failback

Beim automatischen Failback wird eine Ressource automatisch auf den ursprünglichen Node zurückverschoben, sobald dieser wieder verfügbar ist.

Vorteil:

Der gewünschte Normalzustand wird automatisch wiederhergestellt.

Nachteil:

Es kann zusätzliche Bewegung und Last erzeugen.

Beispiel:

Ein Host kommt nach Reparatur zurück und viele VMs wandern automatisch zurück.
Das kann unnötige Last verursachen.

Manuelles Failback

Beim manuellen Failback entscheidet ein Administrator bewusst, wann eine Ressource zurückverschoben wird.

Vorteil:

Kontrollierter Ablauf.

Nachteil:

Mehr manueller Aufwand.

In produktiven Umgebungen ist manuelles Failback oft sinnvoller, weil man vorher prüfen kann, ob der reparierte Node wirklich stabil läuft.

Failover-Voraussetzungen

Damit Failover funktionieren kann, müssen bestimmte Bedingungen erfüllt sein.

Voraussetzung	Erklärung
Mehrere Nodes	Es muss ein Zielsystem geben
Clusterverwaltung	Nodes müssen gemeinsam verwaltet werden
Ressource muss clusterfähig sein	Dienst oder VM muss verschiebbar sein
Zugriff auf Daten	Zielnode muss an die Daten kommen
Netzwerkverbindung	Nodes müssen kommunizieren
Quorum	Cluster muss entscheiden können
Ressourcenreserve	Zielnode muss genug CPU/RAM haben
Monitoring	Fehler müssen erkannt werden

Failover bei VMs mit Shared Storage

Bei Shared Storage greifen mehrere Hosts auf denselben Speicher zu.

Beispiel:

Host	Zugriff auf VM-Daten
HV01	Ja
HV02	Ja
HV03	Ja

Wenn HV01 ausfällt, kann HV02 die VM starten, weil HV02 ebenfalls Zugriff auf die VM-Daten hat.

Vorteil:

Failover ist einfacher, weil die VM-Daten nicht erst kopiert werden müssen.

Failover bei VMs mit lokalem Storage

Wenn eine VM nur auf lokalem Storage eines Hosts liegt, ist Failover schwieriger.

Beispiel:

VM	Speicherort
WEB01	Lokale Disk von HV01

Wenn HV01 ausfällt, kommt HV02 nicht automatisch an die VM-Daten.

Mögliche Lösungen:

Lösung	Erklärung
Replikation	VM-Daten werden auf anderen Host kopiert
Verteilter Storage	Daten liegen verteilt auf mehreren Nodes
Backup-Restore	VM aus Backup wiederherstellen
Shared Storage	Gemeinsamen Speicher nutzen

Failover und Ressourcenreserve

Ein Failover funktioniert nur sinnvoll, wenn der Zielnode genug freie Ressourcen hat.

Beispiel:

Node	Auslastung
HV01	90 %
HV02	95 %
HV03	92 %

Wenn HV01 ausfällt, können HV02 und HV03 die zusätzlichen VMs kaum übernehmen.

Besser:

Node	Auslastung
HV01	60 %
HV02	60 %
HV03	60 %

Dann gibt es Reserven für einen Ausfall.

N+1 beim Failover

N+1 bedeutet, dass eine zusätzliche Reserve vorhanden ist.

Beispiel:

Für die normale Last werden zwei Hosts benötigt.

Es sind drei Hosts vorhanden.

Benötigt	Vorhanden
2 Hosts	3 Hosts

Wenn ein Host ausfällt, können die anderen Hosts die Last übernehmen.

Was ist Quorum?

Quorum ist ein Entscheidungsmechanismus in einem Cluster.

Es legt fest, ob der Cluster oder ein Teil des Clusters weiterarbeiten darf.

Ziel:

Der Cluster soll auch bei Störungen eindeutig entscheiden können.

Quorum verhindert, dass mehrere getrennte Cluster-Teile gleichzeitig aktiv werden.

Merksatz

Quorum sorgt dafür, dass im Cluster nur der Teil weiterarbeitet, der eine gültige Mehrheit besitzt.

Warum braucht ein Cluster Quorum?

Ein Cluster muss erkennen können, ob ein Node wirklich ausgefallen ist oder nur die Verbindung gestört ist.

Beispiel:

Situation	Problem
Node antwortet nicht	Ist der Node aus oder nur nicht erreichbar?
Netzwerk getrennt	Beide Seiten könnten weiterarbeiten wollen
Storage erreichbar	Mehrere Nodes könnten auf dieselben Daten schreiben
Dienst wird doppelt gestartet	Daten können beschädigt werden

Quorum verhindert falsche Mehrfachaktivität.

Quorum und Mehrheit

Viele Cluster arbeiten mit einem Mehrheitsprinzip.

Beispiel mit drei Nodes:

Komponente	Stimme
Node 1	1

Komponente	Stimme
Node 2	1
Node 3	1

Insgesamt gibt es 3 Stimmen.

Mehrheit bedeutet mindestens 2 Stimmen.

Erreichbare Stimmen	Entscheidung
3 von 3	Cluster darf arbeiten
2 von 3	Cluster darf arbeiten
1 von 3	Keine Mehrheit, Cluster sollte stoppen

Quorum-Beispiel mit drei Nodes

Ausgangslage:

Node	Zustand
HV01	Online
HV02	Online
HV03	Online

Alle Nodes haben Verbindung.

Der Cluster hat Quorum.

Jetzt fällt HV03 aus.

Node	Zustand
HV01	Online
HV02	Online
HV03	Offline

HV01 und HV02 haben zusammen noch die Mehrheit.

Der Cluster darf weiterarbeiten.

Quorum-Beispiel mit zwei getrennten Cluster-Teilen

Ein Drei-Node-Cluster wird getrennt.

Cluster-Teil	Nodes	Stimmen
Teil A	HV01, HV02	2
Teil B	HV03	1

Teil A hat die Mehrheit und darf weiterarbeiten.

Teil B hat keine Mehrheit und darf nicht weiterarbeiten.

Dadurch wird verhindert, dass beide Seiten gleichzeitig aktiv sind.

Quorum bei zwei Nodes

Ein Zwei-Node-Cluster ist besonders kritisch.

Beispiel:

Komponente	Stimme
Node 1	1
Node 2	1

Wenn die Verbindung zwischen beiden Nodes ausfällt, hat jede Seite nur eine Stimme.

Ohne zusätzliche Entscheidungshilfe ist unklar, wer weiterarbeiten darf.

Deshalb braucht ein Zwei-Node-Cluster häufig einen Witness.

Witness

Ein Witness ist eine zusätzliche Entscheidungsinstanz für das Quorum.

Er hilft, eine Mehrheit zu bilden.

Beispiele:

Witness-Art	Erklärung
File Share Witness	Netzwerkfreigabe als Zeuge
Disk Witness	Gemeinsamer Datenträger als Zeuge
Cloud Witness	Cloud-Dienst als Zeuge

Merksatz

Ein Witness hilft dem Cluster, bei gerader Node-Anzahl eine eindeutige Mehrheit zu bilden.

Zwei Nodes mit Witness

Beispiel:

Komponente	Stimme
Node 1	1
Node 2	1
Witness	1

Insgesamt gibt es 3 Stimmen.

Mehrheit bedeutet mindestens 2 Stimmen.

Erreichbar	Stimmen	Ergebnis
Node 1 + Node 2 + Witness	3	Cluster arbeitet
Node 1 + Witness	2	Node 1 darf arbeiten
Node 2 + Witness	2	Node 2 darf arbeiten
Nur Node 1	1	Keine Mehrheit
Nur Node 2	1	Keine Mehrheit

File Share Witness

Ein File Share Witness ist eine Netzwerkfreigabe, die als Zeuge für das Quorum verwendet wird.

Beispiel:

Komponente	Beispiel
Witness-Server	FS01
Freigabe	\FS01\ClusterWitness
Aufgabe	Zusätzliche Stimme für Quorum

Vorteil:

Einfach einzurichten.

Nachteil:

Die Freigabe muss zuverlässig erreichbar sein.

Disk Witness

Ein Disk Witness ist ein gemeinsamer Datenträger, der als Zeuge genutzt wird.

Er wird oft in klassischen Shared-Storage-Clustern verwendet.

Vorteil	Nachteil
Gute Integration in Storage-Cluster	Abhängigkeit vom gemeinsamen Storage

---

Cloud Witness

Ein Cloud Witness nutzt einen Cloud-Dienst als Zeuge.

Beispiel:

Ein Windows-Failover-Cluster kann einen Cloud Witness verwenden.

Vorteil:

Nützlich bei verteilten Standorten.

Nachteil:

Abhängigkeit von Internet und Cloudzugang.

Dynamisches Quorum

Manche moderne Cluster können Stimmen dynamisch anpassen.

Ziel:

Der Cluster soll bei Node-Ausfällen möglichst lange entscheidungsfähig bleiben.

Beispiel:

Wenn ein Node geplant heruntergefahren wird, kann der Cluster die Stimmenverteilung anpassen.

Für die IHK reicht meist:

Quorum bedeutet Mehrheitsentscheidung im Cluster.

Was ist Split-Brain?

Split-Brain ist ein gefährlicher Zustand in einem Cluster.

Dabei verlieren Cluster-Teile die Verbindung zueinander und glauben gleichzeitig, aktiv sein zu dürfen.

Beispiel:

Cluster-Teil	Glaubt
Teil A	Ich bin der aktive Cluster
Teil B	Ich bin der aktive Cluster

Wenn beide Teile gleichzeitig auf dieselben Daten schreiben, können Daten beschädigt werden.

Merksatz

Split-Brain bedeutet, dass getrennte Cluster-Teile gleichzeitig aktiv werden und dadurch Daten beschädigen können.

Wie entsteht Split-Brain?

Split-Brain entsteht häufig durch Kommunikationsprobleme.

Beispiele:

Ursache	Erklärung
Netzwerkunterbrechung	Nodes erreichen sich nicht mehr
Heartbeat-Verlust	Cluster erkennt andere Nodes nicht
Falsches Quorum	Keine klare Mehrheitsentscheidung
Fehlender Witness	Besonders kritisch bei zwei Nodes
Storage weiterhin erreichbar	Beide Seiten können auf Daten schreiben
Fehlkonfiguration	Clusterregeln sind falsch gesetzt

Split-Brain-Beispiel

Ein Cluster besteht aus zwei Nodes.

Beide greifen auf denselben Storage zu.

Komponente	Zustand
Node 1	Online

Komponente	Zustand
Node 2	Online
Verbindung zwischen Nodes	Unterbrochen
Shared Storage	Für beide erreichbar

Problem:

Node 1 glaubt, Node 2 sei ausgefallen.
Node 2 glaubt, Node 1 sei ausgefallen.

Wenn beide den Dienst aktiv starten und auf dieselben Daten schreiben, entsteht Datenkorruption.

Warum Split-Brain gefährlich ist

Gefahr	Erklärung
Datenkorruption	Zwei Systeme schreiben gleichzeitig
Inkonsistente Daten	Unterschiedliche Datenstände entstehen
Dienstfehler	Anwendungen arbeiten mit falschem Zustand
Datenverlust	Änderungen können überschrieben werden
Schwierige Wiederherstellung	Manuelle Reparatur nötig
Lange Ausfallzeit	Analyse und Korrektur dauern

Wie verhindert man Split-Brain?

Wichtige Maßnahmen:

Maßnahme	Wirkung
Quorum korrekt planen	Nur Mehrheit darf arbeiten
Witness einsetzen	Hilft bei gerader Node-Anzahl
Redundantes Cluster-Netzwerk	Heartbeat-Ausfall vermeiden
Storage-Zugriff absichern	Nicht mehrere Seiten gleichzeitig schreiben lassen
Fencing / STONITH	Defekten oder isolierten Node ausschalten
Monitoring	Fehler früh erkennen
Tests	Failover-Szenarien prüfen

Fencing

Fencing bedeutet, einen fehlerhaften oder isolierten Node vom Clusterbetrieb auszuschließen.

Ziel:

Ein Node, der nicht mehr sicher koordiniert ist, darf nicht weiter auf gemeinsame Ressourcen zugreifen.

Beispiel:

Situation	Fencing-Reaktion
Node reagiert nicht korrekt	Node wird ausgeschaltet oder isoliert
Node hat Clusterkontakt verloren	Zugriff auf Ressourcen wird verhindert
Gefahr von Split-Brain	Unsicherer Node wird blockiert

STONITH

STONITH steht für:

Shoot The Other Node In The Head

Das klingt hart, bedeutet technisch:

Ein unsicherer Node wird ausgeschaltet oder vom Zugriff ausgeschlossen.

Ziel:

Datenkonsistenz schützen.

Beispiel:

Ein Cluster kann über IPMI, iLO, iDRAC oder PDU einen Node ausschalten, wenn dieser nicht mehr sicher arbeitet.

Warum Fencing wichtig ist

Ohne Fencing könnte ein isolierter Node weiterarbeiten.

Problem:

Auch wenn ein Node keine Clusterverbindung mehr hat, kann er eventuell noch auf Storage schreiben.

Fencing stellt sicher:

Ziel	Erklärung
Nur ein aktiver Zugriff	Daten werden geschützt
Unsicherer Node wird isoliert	Split-Brain wird verhindert
Clusterzustand bleibt eindeutig	Andere Nodes können übernehmen

---

Heartbeat

Heartbeat ist ein regelmäßiges Signal zwischen Cluster-Nodes.

Damit prüfen Nodes gegenseitig, ob sie erreichbar sind.

Beispiel:

Node	Heartbeat
HV01	sendet Signal
HV02	sendet Signal
HV03	sendet Signal

Wenn ein Node keinen Heartbeat mehr sendet, kann der Cluster einen Ausfall vermuten.

Heartbeat-Verlust

Ein Heartbeat-Verlust bedeutet nicht automatisch, dass der Server wirklich ausgefallen ist.

Mögliche Ursachen:

Ursache	Erklärung
Server aus	Echter Ausfall
Netzwerkproblem	Node läuft noch, ist aber nicht erreichbar
Switchproblem	Verbindung unterbrochen
Firewallproblem	Heartbeat blockiert
Überlastung	Node antwortet zu spät

Deshalb darf der Cluster nicht nur nach Gefühl entscheiden.

Quorum und Fencing sind wichtig.

Cluster-Kommunikation

Cluster-Nodes kommunizieren ständig miteinander.

Übertragen werden zum Beispiel:

Information	Bedeutung
Node-Status	Online oder offline
Ressourcenstatus	Welche VM läuft wo?
Heartbeat	Erreichbarkeit
Quorum-Information	Mehrheit vorhanden?
Failover-Ereignisse	Umschaltungen
Konfiguration	Clusterregeln und Zustände

Cluster-Netzwerk redundant auslegen

Das Cluster-Netzwerk sollte nicht nur über eine einzige Verbindung laufen.

Besser:

Maßnahme	Vorteil
Zwei Netzwerkkarten	Schutz bei NIC-Ausfall
Zwei Switches	Schutz bei Switch-Ausfall
Getrennte VLANs	Logische Trennung
Separates Heartbeat-Netz	Weniger Störungen
Monitoring	Fehler schnell erkennen

Failover-Reihenfolge

In manchen Clustern kann man festlegen, auf welchen Node eine Ressource bevorzugt wechseln soll.

Beispiel:

Ressource	Bevorzugter Node	Ersatznode
VM DC01	HV01	HV02
VM FILE01	HV02	HV03
VM WEB01	HV03	HV01

Das hilft bei geordneter Lastverteilung.

Failover-Priorität

Nicht alle VMs sind gleich wichtig.

Beispiel:

VM	Priorität
DC01	Hoch
DNS01	Hoch
DB01	Hoch
FILE01	Mittel
TEST01	Niedrig

Bei Ressourcenknappheit sollten wichtige VMs zuerst gestartet werden.

Testsysteme können später oder gar nicht automatisch starten.

Abhängigkeiten beim Failover

Manche Dienste hängen von anderen Diensten ab.

Beispiel:

Dienst	Benötigt
APP01	DB01
WEB01	APP01
FILE01	DC01 und DNS
BACKUP01	Storage und Netzwerk

Bei Failover oder Neustart muss die Reihenfolge beachtet werden.

Startreihenfolge nach Failover

Eine sinnvolle Startreihenfolge kann sein:

Reihenfolge	System	Grund
1	Netzwerk / Storage	Grundlage
2	Domain Controller / DNS	Anmeldung und Namensauflösung
3	Datenbank	Anwendung benötigt Daten

Reihenfolge	System	Grund
4	Applikationsserver	Anwendung startet
5	Webserver	Zugriff für Benutzer
6	Monitoring / Backup	Kontrolle und Sicherung

Failover-Test

Ein Failover sollte getestet werden.

Warum?

Grund	Erklärung
Funktion prüfen	Failover muss im Ernstfall funktionieren
Ausfallzeit messen	RTO prüfen
Abhängigkeiten erkennen	Startreihenfolge testen
Dokumentation verbessern	Ablauf festhalten
Fehler vor Ernstfall finden	Risiko senken

Geplanter Failover-Test

Typischer Ablauf:

Schritt	Erklärung
Test ankündigen	Benutzer und Verantwortliche informieren
Backup prüfen	Rückfallmöglichkeit sicherstellen
Clusterzustand prüfen	Alle Nodes gesund?
Failover auslösen	Ressource verschieben
Dienst prüfen	Funktioniert der Dienst?
Rückschwenk testen	Failback prüfen
Ergebnis dokumentieren	Erkenntnisse festhalten

Risiken beim Failover-Test

Risiko	Erklärung
Dienstunterbrechung	Benutzer können betroffen sein
Datenbankprobleme	Transaktionen können unterbrochen werden

Risiko	Erklärung
Abhängigkeiten	Folgefehler bei anderen Diensten
Performance	Zielnode kann überlastet werden
Rückfallplan fehlt	Problem dauert länger

Deshalb sollten Failover-Tests geplant und dokumentiert werden.

RTO beim Failover

RTO steht für Recovery Time Objective.

Es beschreibt, wie lange ein Dienst maximal ausfallen darf.

Beispiel:

Dienst	RTO
DNS	15 Minuten
Fileserver	2 Stunden
Testsystem	24 Stunden

Failover hilft, das RTO zu verkürzen.

RPO beim Failover

RPO steht für Recovery Point Objective.

Es beschreibt, wie viele Daten maximal verloren gehen dürfen.

Bei Failover hängt RPO stark von Storage und Replikation ab.

Beispiel:

Technik	Möglicher Datenverlust
Shared Storage	Meist geringer, wenn Storage intakt
Synchrone Replikation	Sehr gering
Asynchrone Replikation	Letzte Änderungen können fehlen
Backup-Restore	Abhängig vom letzten Backup

Failover ersetzt kein Backup

Failover schützt vor Ausfallzeiten, aber nicht vor Datenverlust durch falsche oder beschädigte Daten.

Beispiele:

Problem	Failover hilft?	Backup hilft?
Host fällt aus	Ja	Nicht primär
Datei gelöscht	Nein	Ja
Ransomware verschlüsselt Daten	Nein	Ja, wenn Backup geschützt
Datenbank beschädigt	Nicht zuverlässig	Ja, wenn konsistentes Backup
Fehlkonfiguration	Nicht zuverlässig	Ja, wenn alter Stand vorhanden

Merksatz:

Failover erhöht Verfügbarkeit, Backup schützt vor Datenverlust.

Typische Fehler bei Failover, Quorum und Split-Brain

Fehler	Auswirkung
Kein Witness bei zwei Nodes	Unklare Entscheidung bei Ausfall
Quorum falsch geplant	Cluster kann stoppen oder falsch arbeiten
Kein Fencing	Split-Brain-Gefahr
Cluster-Netzwerk nicht redundant	Heartbeat-Probleme
Shared Storage nicht redundant	Storage bleibt Single Point of Failure
Keine Ressourcenreserve	Failover kann nicht starten
Keine Failover-Tests	Funktion im Ernstfall unsicher
Abhängigkeiten nicht dokumentiert	Dienste starten falsch
Failback automatisch ohne Kontrolle	Unnötige Last oder neue Störungen
Kein Backup	Datenverlust trotz Cluster möglich

Gute Praxis

Empfehlung	Grund
Quorum-Konzept planen	Klare Entscheidungen ermöglichen
Witness bei gerader Node-Anzahl nutzen	Mehrheit herstellen
Fencing einrichten	Split-Brain verhindern

Empfehlung	Grund
Heartbeat-Netz redundant auslegen	Fehlalarme vermeiden
Ressourcenreserve einplanen	Failover ermöglichen
Failover regelmäßig testen	Ernstfall vorbereiten
Failback kontrolliert durchführen	Stabilität prüfen
Abhängigkeiten dokumentieren	Richtige Startreihenfolge
Monitoring einrichten	Probleme früh erkennen
Backups getrennt halten	Datenverlust vermeiden

Prüfungsnahes Beispiel 1

Aufgabe:

Ein Zwei-Node-Cluster verliert die Netzwerkverbindung zwischen den beiden Nodes. Beide Nodes können weiterhin auf den gemeinsamen Storage zugreifen.

Frage:

Welches Risiko entsteht?

Mögliche Antwort:

Es besteht die Gefahr eines Split-Brain-Zustands. Beide Nodes könnten glauben, dass der jeweils andere ausgefallen ist und gleichzeitig aktiv werden. Wenn beide auf denselben Storage schreiben, kann es zu Datenkorruption kommen. Ein korrektes Quorum mit Witness und Fencing soll das verhindern.

Prüfungsnahes Beispiel 2

Aufgabe:

Ein Cluster besteht aus drei Nodes. Ein Node fällt aus.

Frage:

Darf der Cluster weiterarbeiten?

Mögliche Antwort:

Ja, wenn die verbleibenden zwei Nodes noch miteinander kommunizieren können. Bei drei Stimmen besitzen zwei erreichbare Nodes weiterhin die Mehrheit. Das Quorum ist vorhanden.

Prüfungsnahes Beispiel 3

Aufgabe:

Ein Cluster besteht aus zwei Nodes ohne Witness. Die Verbindung zwischen den Nodes wird unterbrochen.

Frage:

Warum ist das problematisch?

Mögliche Antwort:

Ohne Witness gibt es keine eindeutige Mehrheit. Jeder Node hat nur seine eigene Stimme. Dadurch kann der Cluster nicht sicher entscheiden, welcher Node weiterarbeiten darf. Es besteht Risiko für Dienstunterbrechung oder Split-Brain, je nach Konfiguration.

Prüfungsnahes Beispiel 4

Aufgabe:

Ein Administrator möchte einen Host warten und verschiebt vorher die VMs auf einen anderen Host.

Frage:

Ist das Failover oder Live Migration?

Mögliche Antwort:

Das ist eine geplante Verschiebung und wird als Live Migration bezeichnet, wenn die VMs dabei weiterlaufen. Failover beschreibt eher die Übernahme bei einem Fehler oder Ausfall.

Prüfungsnahes Beispiel 5

Aufgabe:

Ein Unternehmen hat einen Cluster, aber keine Backups. Ein Benutzer löscht versehentlich wichtige Daten.

Frage:

Warum hilft der Cluster nicht?

Mögliche Antwort:

Der Cluster sorgt für Verfügbarkeit, aber er stellt keine alten Datenstände wieder her. Wenn Daten gelöscht werden, ist der gelöschte Zustand auch im aktiven Dienst vorhanden. Für die Wiederherstellung wird ein Backup benötigt.

Typische Prüfungsfrage: Was ist Failover?

Mögliche Antwort:

Failover bedeutet, dass ein Dienst oder eine Ressource bei Ausfall eines Systems auf ein anderes System wechselt oder dort neu gestartet wird. Dadurch wird die Ausfallzeit reduziert.

Typische Prüfungsfrage: Was ist Failback?

Mögliche Antwort:

Failback bedeutet, dass eine Ressource nach der Reparatur des ursprünglichen Systems wieder zurück auf dieses System verschoben wird.

Typische Prüfungsfrage: Was ist Quorum?

Mögliche Antwort:

Quorum ist ein Entscheidungsmechanismus in einem Cluster. Es stellt sicher, dass nur der Cluster-Teil mit gültiger Mehrheit weiterarbeiten darf. Dadurch werden gefährliche Mehrfachaktivitäten verhindert.

Typische Prüfungsfrage: Warum braucht ein Zwei-Node-Cluster oft einen Witness?

Mögliche Antwort:

Bei zwei Nodes gibt es ohne Witness keine klare Mehrheit, wenn die Verbindung zwischen den Nodes unterbrochen wird. Ein Witness liefert eine zusätzliche Stimme und hilft dem Cluster, eindeutig zu entscheiden, welcher Teil weiterarbeiten darf.

Typische Prüfungsfrage: Was ist Split-Brain?

Mögliche Antwort:

Split-Brain ist ein gefährlicher Cluster-Zustand, bei dem getrennte Cluster-Teile gleichzeitig aktiv werden. Wenn beide Seiten auf dieselben Daten schreiben, kann es zu Datenkorruption oder Datenverlust kommen.

Typische Prüfungsfrage: Wie verhindert man Split-Brain?

Mögliche Antwort:

Split-Brain wird durch ein korrekt geplantes Quorum, einen Witness bei Bedarf, redundante Cluster-Kommunikation und Fencing verhindert. Dadurch darf nur der berechtigte Cluster-Teil weiterarbeiten.

Typische Prüfungsfrage: Was ist Fencing?

Mögliche Antwort:

Fencing bedeutet, einen unsicheren oder isolierten Node vom Zugriff auf gemeinsame Ressourcen auszuschließen. Dadurch wird verhindert, dass dieser Node weiterarbeitet und Daten beschädigt.

Typische Prüfungsfrage: Was bedeutet STONITH?

Mögliche Antwort:

STONITH steht für „Shoot The Other Node In The Head“. Gemeint ist, dass ein unsicherer Node ausgeschaltet oder isoliert wird, damit er keine gemeinsamen Daten beschädigen kann.

Typische Prüfungsfrage: Unterschied Failover und Live Migration

Failover	Live Migration
Reaktion auf Fehler oder Ausfall	Geplante Verschiebung
Häufig kurze Unterbrechung	Möglichst ohne Unterbrechung
Beispiel: Host fällt aus	Beispiel: Host wird gewartet
Cluster startet Ressource woanders	Laufende VM wird verschoben

---

Typische Prüfungsfrage: Warum braucht Failover Ressourcenreserven?

Mögliche Antwort:

Wenn ein Node ausfällt, müssen die verbleibenden Nodes dessen Dienste oder VMs übernehmen. Sind die verbleibenden Nodes bereits vollständig ausgelastet, kann das Failover nicht zuverlässig funktionieren. Deshalb braucht ein Cluster freie CPU-, RAM-, Storage- und Netzwerkressourcen.

Typische Prüfungsfrage: Warum ersetzt Failover kein Backup?

Mögliche Antwort:

Failover reduziert Ausfallzeiten bei Systemfehlern, stellt aber keine alten Datenstände wieder her. Bei gelöschten, verschlüsselten oder beschädigten Daten wird weiterhin ein Backup benötigt.

Wichtige Begriffe

Begriff	Kurz erklärt
Failover	Übernahme bei Ausfall
Failback	Rückkehr auf ursprüngliches System
Automatisches Failover	Cluster reagiert selbst auf Fehler
Manuelles Failover	Administrator löst Umschaltung aus
Geplantes Failover	Bewusste Umschaltung für Wartung oder Test
Ungeplantes Failover	Umschaltung durch plötzlichen Fehler
Quorum	Entscheidungsmechanismus im Cluster
Mehrheit	Genug Stimmen für Weiterbetrieb
Witness	Zusätzliche Entscheidungsinstanz
File Share Witness	Netzwerkfreigabe als Zeuge
Disk Witness	Gemeinsamer Datenträger als Zeuge
Cloud Witness	Cloud-Dienst als Zeuge
Split-Brain	Getrennte Cluster-Teile werden gleichzeitig aktiv
Heartbeat	Regelmäßiges Signal zwischen Nodes
Fencing	Unsicheren Node ausschließen
STONITH	Unsicheren Node ausschalten oder isolieren
RTO	Maximal erlaubte Ausfallzeit
RPO	Maximal erlaubter Datenverlust
Live Migration	Laufende VM geplant verschieben
Ressourcenreserve	Freie Kapazität für Failover

Wichtige Merksätze

Failover bedeutet Übernahme bei Ausfall.

Failback bedeutet Rückkehr auf das ursprüngliche System.

Quorum sorgt für eine eindeutige Entscheidung im Cluster.

Ein Witness hilft besonders bei Clustern mit gerader Node-Anzahl.

Split-Brain ist ein gefährlicher Zustand, bei dem getrennte Cluster-Teile gleichzeitig aktiv werden.

Fencing verhindert, dass ein unsicherer Node weiter auf gemeinsame Ressourcen zugreift.

Heartbeat zeigt Erreichbarkeit, beweist aber nicht allein den echten Zustand eines Servers.

Failover braucht Ressourcenreserven.

Failover reduziert Ausfallzeiten, ersetzt aber kein Backup.

Live Migration ist geplant, Failover ist meist eine Reaktion auf einen Fehler.

Kurzzusammenfassung

Failover, Quorum und Split-Brain gehören zu den wichtigsten Grundlagen beim Verständnis von Clustern.

Thema	Kerngedanke
Failover	Ein Dienst oder eine VM wird bei Ausfall auf einem anderen Node gestartet
Failback	Nach Reparatur wird auf das ursprüngliche System zurückgeschwenkt
Quorum	Der Cluster entscheidet über eine Mehrheit
Witness	Zusätzliche Stimme zur Mehrheitsbildung
Split-Brain	Gefährliche Mehrfachaktivität getrennter Cluster-Teile
Heartbeat	Regelmäßige Erreichbarkeitsprüfung
Fencing	Unsichere Nodes werden ausgeschlossen
STONITH	Unsichere Nodes werden ausgeschaltet oder isoliert

Für die IHK ist besonders wichtig:

Prüfungsrelevanter Punkt	Bedeutung
Failover erklären	Übernahme bei Ausfall
Quorum erklären	Mehrheitsentscheidung im Cluster
Witness kennen	Zusätzliche Stimme bei gerader Node-Anzahl
Split-Brain verstehen	Gefahr für Datenkonsistenz
Fencing verstehen	Schutz vor unsicheren Nodes

Prüfungsrelevanter Punkt	Bedeutung
Failover vs. Live Migration unterscheiden	Fehlerfall vs. geplante Verschiebung
Cluster vs. Backup unterscheiden	Verfügbarkeit ist keine Datensicherung

Der nächste logische Schritt ist:

Load Balancing und Skalierung
