

# 1&1 IONOS Enterprise Cloud

## Automatisiertes Setup für Gateway- und Datenbankcluster

In diesem Text wird der Anlass und der allgemeine Hintergrund des Programms beschrieben. Die Installation wird in der README.md erläutert.

Dieses Projekt soll ein vollwertiges Setup mit Gateways und Datenbankservern im Hot-Standby-Modus in einem virtuellen Rechenzentrum der Enterprise Cloud von 1&1 IONOS mit Hilfe eines einzigen Scriptes ermöglichen.

Um seinen Nutzen und die Vergleichbarkeit zu belegen ist es möglich, die produktive Datenbank [MusicBrainz](#) zu importieren.

### Grundlegende Schritte

1. Zunächst wird ein virtuelles Rechenzentrum (virtual data center, VDC) in der Enterprise Cloud erstellt und drei Kunden-IP-Adressen reserviert. Dies ist notwendig, um sicherzustellen, dass eine Adresse nach einem Neustart der virtuellen Maschine (VM) nicht verloren geht.
2. Im nächsten Schritt wird eine Gruppe von zwei VMs erstellt, welche die Verbindung zum Internet über eine virtuellen IP sicher stellen, die bei einem Ausfall der Master-VM von der Standby-VM übernommen werden kann.
3. Beim Erstellen der Gateway-VMs wird die Failover-Gruppe erstellt. Damit wird gewährleistet, dass der gesamte Internetverkehr von Providerseite an die aktive VM gesendet wird, der die externe Failover-IP zugeordnet ist.
4. Anschließend wird eine VM als Management-Instanz angelegt, um ein Beispiel für die Erreichbarkeit des internen Netzwerks abzubilden. Das erste Ziel des Projekts war es, genau das zu demonstrieren. In dieser VM wird SaltStack installiert, um damit die nächsten Schritte der Installation durchführen zu lassen. Es ist auch möglich, das Setup an dieser Stelle zu beenden. Somit ergäbe sich ein VDC, mit welchem die Funktion der Failover-Gruppe und der virtuellen IP extern und innerhalb des privaten Netzwerks testen ließe. (Eine detailliertere Beschreibung der Failover-Konfiguration finden sich in der Artikelreihe: [‘The Failover Cloud Solution Part 1 to 3’](#) des 1&1 IONOS Professional Services Consultants Thomas Vogel.)

Wenn dies alles ist, was benötigt wird, muss das Skript einfach ausschließlich mit der Option '-l' und der Angabe des Ortes, z.B. 'fra' für Frankfurt gestartet werden. Wird auch der Datenbankcluster benötigt, kommen die nächsten zwei Absätze ins Spiel.

5. Die nächste Aufgabe ist das Einrichten von zwei PostgreSQL-Instanzen in einer Master-Standby-Konfiguration. Der Cluster wird über eine von PGPool verwaltete Virtual Private IP erreichbar sein, über die auch die Anfragen an die Datenbank geleitet werden. Die interne virtuelle IP ist von extern über eine Portweiterleitung erreichbar, die von den Gateway-VMs

bereitgestellt wird. Zum Testen und Vergleichen der Performance können verschiedene zugrunde liegende Dateisysteme ausgewählt werden.

6. Als letzte Aufgabe wird noch eine aktuelle Ausgabe der Datenbank [MusicBrainz](#) in das Setup geladen.

Um zu starten ist der Download des Programmpakets durchzuführen, es zu entpacken, in das erstellte Unterverzeichnis zu wechseln und 'HA-GateAndDatabase.sh' auszuführen. Das Starten des Skripts ohne Optionen zeigt die Verwendung des Programms an:

<snip>

```
Usage: ./HA-GateAndDatabase.sh -l location [-d] [-D] [-M] [-s] [-i] [-f  
filesystem] [-S number]
```

Location can be fra (Frankfurt), fkb (Karlsruhe), ewr (Newark) and las (Las Vegas)

-d activate debug mode

-D install the database cluster

-M install the MusicBrainz database

The filesystem for the DB would be expanded to at least 300 GB

-s use SSD for database volumes

-i use Intel for database VM cores

-f filesystem

The filesystem could be ext4 (default), xfs, btrfs, zfs

-S size of database volume in GB

Just "./HA-GateAndDatabase.sh -l location" installs only the gateway HA setup with the management host behind it.

The user credentials need to be placed in a file named /home/gschiechedirik/ionos/.config in the form 'user@domain.tdl:password' (without apostrophes).

With the "-d" option, your Data Center Designer password will be shown on the console!

</snip>

Mit dem Kommando

```
'./HA-GateAndDatabase.sh -l fra'
```

wird ein simples High-Availability-Setup im IONOS Rechenzentrum in Frankfurt erstellt. Um die Funktionalität einer Failovergruppe im Data Center Designer und innerhalb der VMs kennen zu lernen ist dies schon ausreichend.

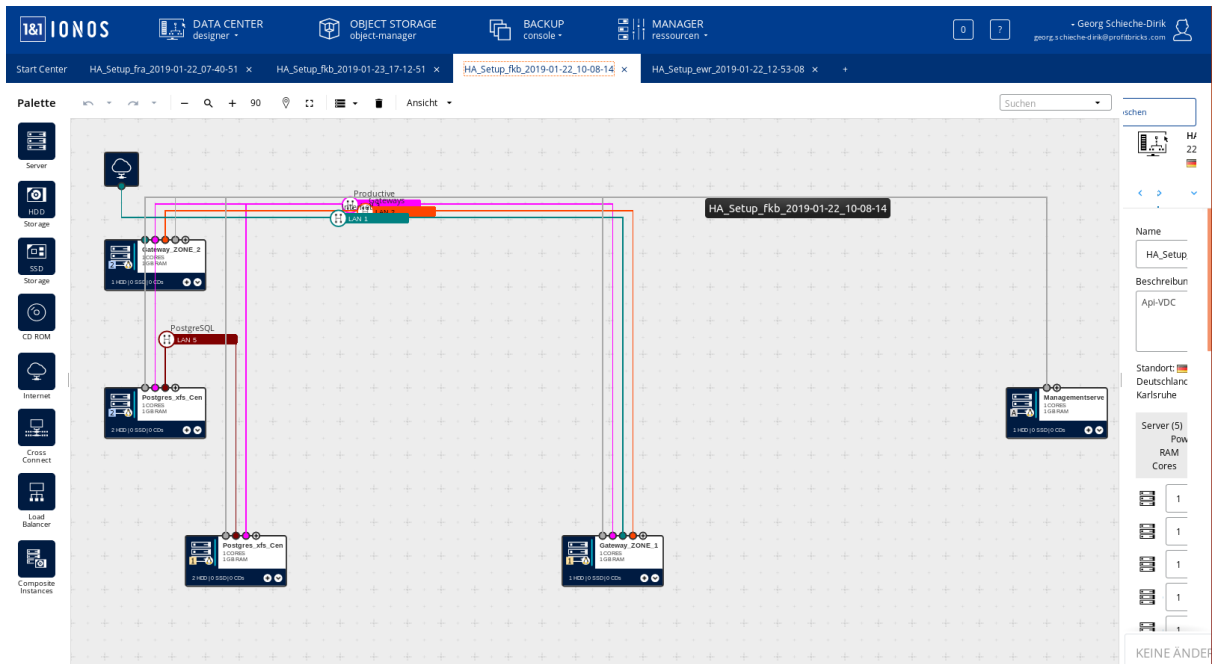
Um auch den Datenbankcluster zu testen ist das folgende Kommando auszuführen:

```
'./HA-GateAndDatabase.sh -l fra -D -M -f xfs'
```

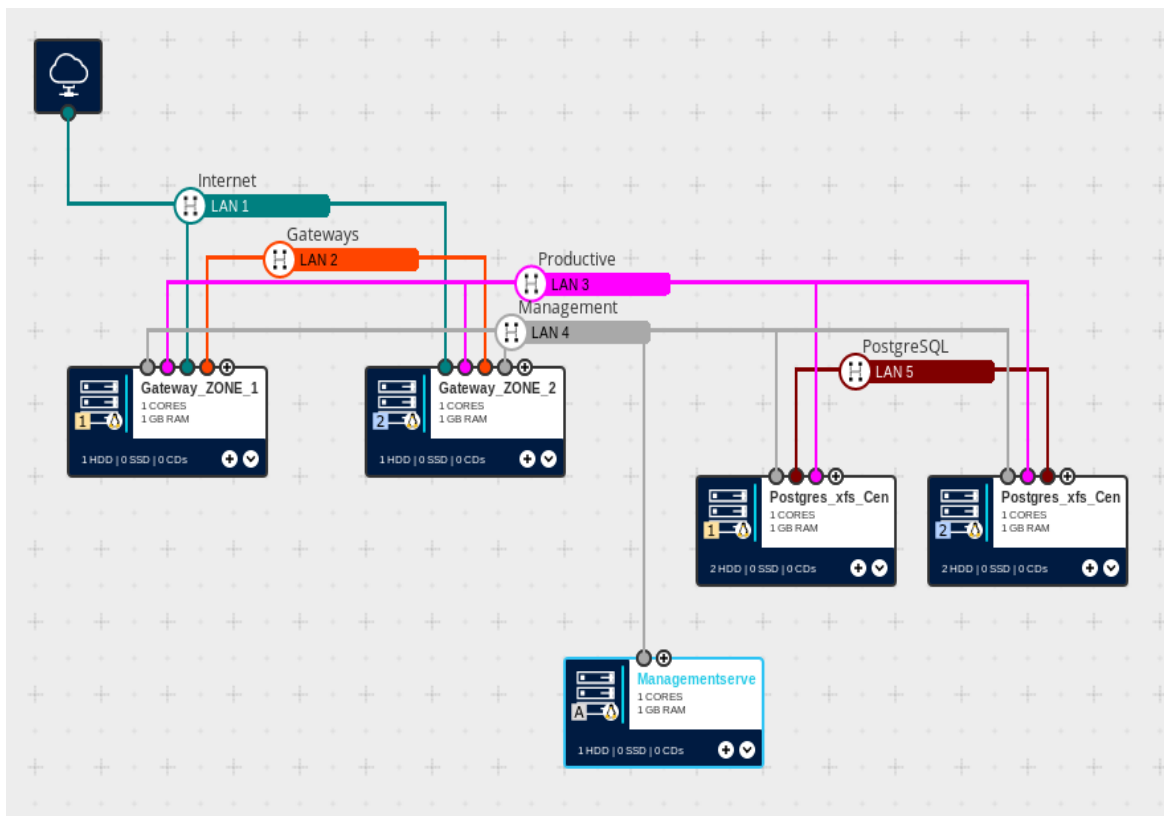
Das Script erzeugt jetzt die Gateway-Failovergruppe und den ebenfalls hochverfügbaren Datenbankcluster auf dem Dateisystem XFS.

# IP-Failover Setup im Data Center Designer

Wenn das Skript durchgelaufen ist, sieht das virtuelle Rechenzentrum zunächst so aus.



Nach dem funktionalen Anordnen der VMs ist die Struktur sichtbar.



Die Failovergruppe im DCD ist für die VMs Gateway\_ZONE\_1 und Gateway\_ZONE\_2 konfiguriert.

The screenshot displays the Data Center Designer interface. On the left, a network diagram shows a cloud icon connected to LAN 1, which is part of a 'Gateways' group. LAN 2 is connected to 'Productive Management' (LAN 3). Two VMs, Gateway\_ZONE\_1 and Gateway\_ZONE\_2, are shown with their respective network connections. Gateway\_ZONE\_1 is connected to LAN 1 and LAN 2, while Gateway\_ZONE\_2 is connected to LAN 2 and LAN 3. On the right, the 'Gateway\_ZONE\_1' configuration panel is open, showing the 'Netzwerk' (Network) tab. It lists three NICs: NIC 0, NIC 1, and NIC 2. NIC 1 is highlighted in pink. The configuration for NIC 2 is shown below, with fields for Name, MAC, Primary IP (46.16.76.43), LAN (LAN 1), Firewall, and DHCP. The DHCP section is checked, showing a Gateway IP of 46.16.76.1 and an additional IP of 46.16.74.47 marked as 'Failover'.

Auf der rechten Seite in der Ansicht des Data Center Designers ist die externe NIC-Konfiguration des Gateway\_ZONE\_1 dargestellt.

Die primäre externe IP ist 46.16.76.43. Die **virtuelle IP** ist die zusätzliche IP **46.16.74.47**. **Diese IP wird für den Failover verwendet.**

Das Konfigurationsmenu für die Failover-Gruppe.

The screenshot shows the 'IP-Failover' configuration menu. At the top, there is a section for 'LAN 1: Internet' with a public status and options for color and actions. Below this, the 'Member' tab is selected, showing a list of IP addresses. The first IP address is 46.16.74.47, which is the virtual IP. Below the IP address, there are options to 'Verwalten' (Manage), 'Ändern' (Change), and 'Löschen' (Delete). The 'NIC' section is also visible, showing a list of NICs. The first NIC is 'NIC 0 an "Gateway\_ZONE\_2"', which is currently selected. The second NIC is 'NIC 2 an "Gateway\_ZONE\_1"', which is currently unselected. The 'Master\*' status is indicated for the selected NIC.

## Intention des Vorhabens:

1. Zuerst ging es darum die DCD API kennenlernen, um jederzeit in der Lage zu sein, jede Funktion der Programmierschnittstelle anzusprechen und zu überprüfen. Es gibt mehrere andere Tools oder SDKs, die diesen Job erledigen, aber wenn etwas nicht wie erwartet funktioniert, ist es nie sicher, ob die API ein Problem hat oder die verwendete Software. Mit Shell-Befehlen, beispielsweise 'curl', haben User die Möglichkeit, die API ohne Zwischenschicht anzusprechen.
2. Wird das Skript mit der Debug-Option '-d' gestartet, dann wird jeder Aufruf explizit angezeigt. Dadurch ist es einfach, für weitere Aufrufe die richtige Syntax zu finden und zu überprüfen, ob alle verwendeten Funktionen wie erwartet arbeiten. Damit ist es sehr einfach, die Schritte als Vorlage für andere Projekte zu verwenden.
3. Das Virtual Router Redundancy Protocol (VRRP) wird häufig für ein HA-Setup mit einer virtuellen IP verwendet, insbesondere in unserer Infrastruktur. Daher ist es sinnvoll, es für dieses Setup zu verwenden. Nach dem Aufruf des Skripts wird das gesamte Setup mit Shell-Befehlen durchgeführt. Die notwendigen Firewall-Einstellungen werden ebenfalls vorgenommen.
4. In der Management-VM wird SaltStack installiert. Mit diesem Tool werden große Netzwerke von Computern gesteuert. Damit ist es ein Kandidat für die Konfiguration der Datenbank-VMs und die Installation aller notwendigen Anwendungen.
5. Die PostgreSQL-Datenbank ist als Master-Standby-Cluster konfiguriert. Der Standby wird durch Streaming-Replikation synchron gehalten. Der Standby kann Leseanfragen bearbeiten und die Masterrolle übernehmen, falls der Master nicht mehr arbeitet. Wenn der ehemalige Master neu gestartet wird und wieder online geht, lässt die Salt Konfiguration ihn zum neuen Standby werden (während der vormalige Standby die Masterrolle behält). Als Dateisystem kann zwischen ext4, btrfs, xfs und zfs gewählt werden.
6. Die Verteilung von Lese- und Schreib Anforderungen für die Datenbank und die Handhabung der virtuellen IP für die produktive Schnittstelle wird von PGPool verwaltet. Dies wird auch von SaltStack konfiguriert.
7. Als Beispiel für eine "komplette produktive Datenbank" wird die Datenbank MusicBrainz (siehe [https://musicbrainz.org/doc/About/Data\\_License](https://musicbrainz.org/doc/About/Data_License)) installiert. Diese Installation wird ebenfalls von SaltStack durchgeführt. Der Teil des Imports von MusicBrainz dauert zwischen vier und fünf Stunden. Erst danach (!) sollte das gesamte Failover-Setup getestet werden. Das Recovern einer Instanz nach einem Ausfall dauert auch mindestens eine Stunde, bis die Datenbank wieder synchronisiert ist.
8. Ein einfacher Test des Setups besteht darin, die Master-Instanz einfrieren zu lassen und zu überprüfen, ob das Failover wie erwartet funktioniert.
9. Mit 'ssh -p 22222 root@46.16.74.61' (als ein Beispiel für eine externe IP) wird eine direkte Verbindung zur Management-VM gestartet. Der Port 22222 des Gateways wird an den Port 22 dieser VM weiter geleitet. Ebenso wird der Port 25432 auf den port 9999 der virtuellen

IP von PGPool weitergeführt, so dass beispielsweise mit  
'ssh -L 33333:127.0.0.1:9999 -p 25432 root@46.16.74.61' ein Tunnel zur Datenbank  
eingerrichtet wird. Fällt nun der Master der Datenbank aus, wird PGPool die virtuelle IP  
automatisch an den Standby übergeben, der nun zum Master wird. Der Tunnel muss dann  
neu erstellt werden und die Verbindung steht wieder bereit.k

Mit dieser Applikation verfügen wir über ein hochverfügbares Datenbank-Setup, mit dem neue  
Software, deren Leistung auf unserer Plattform und neue Konfigurationen überprüft und getestet  
werden können.

Alle genannten Features in einem Script zusammengefasst und mit einem Aufruf realisiert, machen  
es außerordentlich schnell und leicht möglich, komplexe Test durchzuführen ohne viel Zeit in den  
Aufbau investieren zu müssen.