SQL Server Hochverfügbar-keitsgruppen mit Ubuntu

Kathegorie
SQL Server

Simon
Schlaipfer

In diesem Beitrag wollen wir uns einmal anschauen, wie sich mithilfe von **Pacemaker Cluster** eine SQL Server Hochverfügbarkeitsgruppe auf Ubuntu 18.04 konfigurieren lässt. Hierfür erstellen wir ein Cluster aus 3 Knoten, wobei jeder Knoten ein eigener Ubuntu 18.04 Server ist.

Zur Übersicht hier einmal die Nodes mit ihrem Computernamen und ihrer virtuellen IP:

- **UBSQL01** - 172.17.1.69
- **UBSQL02** - 172.17.1.72
- **UBSQL03** - 172.17.1.74

Nachdem wir unsere 3 Ubuntu Maschinen erfolgreich erstellt haben, wollen wir als Erstes überprüfen, ob wir uns von UBSQL01 mit UBSQL02 und UBSQL03 via **ssh** verbinden können. Hierfür loggen wir uns zunächst auf UBSQL01 ein und verbinden uns dann mit dem folgenden Befehl mit UBSQL02:

```
ssh 172.17.1.72
```

Mit **exit** kann die Verbindung anschließend wieder getrennt werden. Nun versuchen wir uns noch mit UBSQL03 zu verbinden und wiederholen diesen Vorgang anschließend mit UBSQL02 und UBSQL03.

Waren alle Verbindungen erfolgreich, können wir mit dem eigentlichen Setup der Hochverfügbarkeitsgruppe beginnen.

SQL Server installieren

Die hier aufgeführten Schritte müssen auf jeder Node einzeln durchgeführt werden!

Als ersten Schritt auf dem Weg zu unserer Hochverfügbarkeitsgruppe müssen wir auf all unseren Nodes SQL Server installieren. Hierfür müssen wir als Erstes einen Registrierungsschlüssel importieren. Diesen können wir auf Ubuntu mit dem folgenden Befehl erhalten:

```
sudo wget -qO- https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
```


Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Nachdem wir die Keys importiert haben, muss noch ein Microsoft SQL Server Ubuntu Repository hinzugefügt werden. Dies können wir mit diesem Befehl hinzufügen:

```
sudo add-apt-repository "$(wget -qO- https://packages.microsoft.com/config/
ubuntu/18.04/mssql-server-2019.list)"
```

Waren diese Schritte erfolgreich, können wir mit den folgenden Befehlen das SQL Server Package installieren:

```
sudo apt-get update
sudo apt-get install -y mssql-server
```

Nachdem das Package installiert ist, kann das SQL Server Configuration Skript zum Aufsetzen einer Instanz verwendet werden. Das Skript kann mit diesem Kommando ausgeführt werden:

```
sudo /opt/mssql/bin/mssql-conf setup
```

Hier muss nun eine SQL Server Edition angeben und die Nutzungsbedingungen akzeptiert werden. Anschließend muss noch ein SA Passwort vergeben werden.

Nach der erfolgreichen Installation von SQL Server, müssen wir noch die SQL Server command-line tools installieren. Um die SQL Server command-line tools zu installieren, kann folgendes Skript ausgeführt werden:

```
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
curl https://packages.microsoft.com/config/ubuntu/18.04/prod.list | sudo tee /etc/
apt/sources.list.d/msprod.list
sudo apt-get update
sudo apt-get install mssql-tools unixodbc-dev
```

Zeile 1 Importiert die GPG-Schlüssel des öffentlichen Repositories
Zeile 2 Registriert das Microsoft Ubuntu-Repository
Zeile 3, 4 Führt den Installationsprozess aus

Wenn die Kommandos richtig ausgeführt wurden, erscheint eine Aufforderung zum Annehmen der Lizenzbedingungen. Diese müssen akzeptiert werden, um den Vorgang abzuschließen.

Nun müssen wir SQL Server HA (high availability) installieren, um SQL Server für die Hochverfügbarkeitsgruppe zu konfigurieren. Dies tun wir mit diesem Kommando:

```
sudo apt-get install mssql-server-ha
```

SQL Server HA kann nun mit folgendem Kommando aktiviert werden:

```
sudo /opt/mssql/bin/mssql-conf set hadr.hadrenabled  1
```

Anschließend muss SQL Server mit diesem Kommando neugestartet werden:

```
sudo systemctl restart mssql-server
```

Nachdem wir diese Schritte auf allen Nodes durchgeführt haben, können wir mit der Installation von **Pacemaker** beginnen.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Pacemaker Agenten installieren und das Cluster konfigurieren

Pacemaker ist ein open-source Hochverfügbarkeitsgruppen Ressourcen-Manager, der Fehler auf einzelnen Nodes erkennen und Ressourcen zwischen einzelnen Nodes verschieben kann.

Um den Pacemaker zu installieren, führen wir auf allen drei Nodes dieses Kommando aus:

```
sudo apt-get install pacemaker pcs fence-agents resource-agents
```

Als Nächstes müssen wir die benötigen Ports öffnen. Auch dies muss auf allen drei Nodes durchgeführt werden:

```
sudo ufw allow 2224/tcp
sudo ufw allow 3121/tcp
sudo ufw allow 21064/tcp
sudo ufw allow 5405/udp
sudo ufw allow 1433/tcp
sudo ufw allow 5022/tcp
sudo ufw reload
```

Das letzte Kommando aktualisiert einmal die Firewall.

Anschließend muss ein Passwort für den Benutzer **hacluster** festgelegt werden. Dieser Benutzer authentifiziert andere Nodes innerhalb des Pacemaker Clusters. Dies kann mit diesem Kommando durchgeführt werden:

```
sudo passwd hacluster
```

Nun muss Pacemaker auf allen Nodes mit dem folgenden Kommando aktiviert und gestartet werden:

```
sudo systemctl enable pcsd
sudo systemctl start pcsd
sudo systemctl enable pacemaker
```

Zur Sicherheit entfernen wir alle vorherigen Cluster:

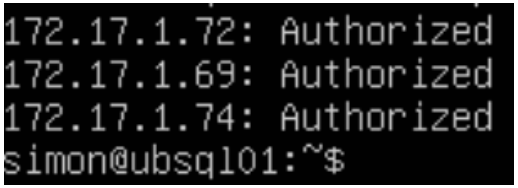
```
sudo pcs cluster destroy
sudo systemctl enable pacemaker
```

Jetzt können wir die einzelnen Nodes zu einer Hochverfügbarkeitsgruppe zusammenführen. Hierfür müssen wir die einzelnen Nodes von der primären Node aus authentifizieren. Hierfür führen wir **nur** auf der Node, die wir als primäre Node festlegen wollen (in unserem Falle UBSQL01), folgendes Kommando aus:

```
sudo pcs cluster auth 172.17.1.69 172.17.1.72 172.17.1.74 -u hacluster -p
admin@123
```

Für den Parameter -p geben wir das Passwort ein, welches wir vorher für den User **hacluster** festgelegt haben.

War das authentifizieren erfolgreich, erscheint folgende Ausgabe:



Nach erfolgreicher Authentifizierung können wir das Pacemaker Cluster erstellen. Dafür führen wir auf unserer primären Node, UBSQL01, folgendes Kommando aus:

```
sudo pcs cluster setup --name ubag 172.17.1.69 172.17.1.72 172.17.1.74
sudo pcs cluster start --all
sudo pcs cluster enable --all
```

War das Erstellen, Starten und Aktivieren des Clusters erfolgreich, erscheint eine vergleichbare Ausgabe:

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

```
Synchronizing pcsd certificates on nodes 172.17.1.69, 172.17.1.72, 172.17.1.74...
172.17.1.69: Success
172.17.1.74: Success
172.17.1.72: Success
Restarting pcsd on the nodes in order to reload the certificates...
172.17.1.72: Success
172.17.1.74: Success
172.17.1.69: Success
simon@ubsql01:~$ sudo pcs cluster start --all
172.17.1.72: Starting Cluster...
172.17.1.69: Starting Cluster...
172.17.1.74: Starting Cluster...
simon@ubsql01:~$ sudo pcs cluster enable --all
172.17.1.69: Cluster Enabled
172.17.1.72: Cluster Enabled
172.17.1.74: Cluster Enabled
```

Abschließend müssen wir einen SQL Server Login für Pacemaker erstellen. Das Pacemaker Cluster nutzt diesen Login, um sich mit SQL Server zu verbinden und Änderungen an der Hochverfügbarkeitsgruppe durchzuführen. Dafür muss auf allen drei Nodes das T-SQL Skript ausgeführt werden:

```
CREATE LOGIN [pacemakerLogin] with PASSWORD= N'admin@123'
ALTER SERVER ROLE [sysadmin] ADD MEMBER [pacemakerLogin]
GO
```

Den Login und das Passwort speichern wir nun im Ordner **pacemaker-passwd** und speichern diesen in **/var/opt/mssql/secrets/passwd**. Hierfür bitte folgendes Skript ausführen:

```
echo 'pacemakerLogin' >> ~/pacemaker-passwd
echo 'admin@123' >> ~/pacemaker-passwd
sudo mv ~/pacemaker-passwd /var/opt/mssql/secrets/passwd
sudo chown root:root /var/opt/mssql/secrets/passwd
sudo chmod 400 /var/opt/mssql/secrets/passwd
```

Die Hochverfügbarkeitsgruppe erstellen

Um nun die Hochverfügbarkeitsgruppe zu erstellen, verbinden wir uns mit SQL Server auf unserer primären Node und erstellen einen sogenannten **Master Key** mit folgendem T-SQL Skipt:

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Cert@123A'
GO
CREATE CERTIFICATE AGcert with subject ='dbm'
GO
```

Um ein Backup für den **Master Key** zu erstellen, folgendes T-SQL Skript ausführen:

```
Backup certificate AGcert
to file ='/var/opt/mssql/data/AGcert.cer'
With private key (
File = '/var/opt/mssql/data/AGcert.pvk',
Encryption by password = 'Cert@123A'
)
GO
```

Die Backup-Dateien müssen anschließend auf die anderen Nodes nach **/var/opt/mssql/data** kopiert werden.
Mit diesen Backup-Dateien können wir dann auf den anderen Nodes ein Zertifikat erstellen. Hierfür muss sichergestellt werden, dass der Benutzer **mssql** die Berechtigung zum Erstellen eines Zertifikat besitzt. Zum Erstellen des Zertifikats bitte folgendes T-SQL Skript ausführen:

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Cert@123A';
GO
CREATE CERTIFICATE AGcert
from file ='/var/opt/mssql/data/AGcert.cer'
With private key (
File = '/var/opt/mssql/data/AGcert.pvk',
decryption by password = 'Cert@123A'
)
GO
```


Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Als Nächstes muss ein Endpunkt für alle drei Nodes mittels eines AGcert Zertifikats erstellt werden. Hierfür muss auf jeder Node dieses T-SQL Skript ausgeführt werden:

```
create endpoint [AG_ENDPOINT]
as TCP (LISTENER_PORT = 5022)
FOR DATABASE_MIRRORING (
ROLE =ALL,
AUTHENTICATION = CERTIFICATE AGcert,
encryption = required algorithm aes
)
GO
ALTER ENDPOINT [AG_ENDPOINT] STATE =STARTED
GO
```

Nun kann eine Hochverfügbarkeitsguppe erstellt werden. Dafür folgendes Skript ausführen:

```
CREATE AVAILABILITY GROUP [ubag]
with (DB_FAILOVER =ON ,CLUSTER_TYPE=EXTERNAL)
FOR REPLICA ON
N'172.17.1.69'
WITH
(ENDPOINT_URL = N'tcp://UBSQL01:5022',
AVAILABILITY_MODE =SYNCHRONOUS_COMMIT,
FAILOVER_MODE =EXTERNAL,
SEEDING_MODE =AUTOMATIC
),
N'172.17.1.72'
WITH
(ENDPOINT_URL = N'tcp://UBSQL02:5022',
AVAILABILITY_MODE =SYNCHRONOUS_COMMIT,
FAILOVER_MODE =EXTERNAL,
SEEDING_MODE =AUTOMATIC
),
N'172.17.1.74'
WITH
(ENDPOINT_URL = N'tcp://UBSQL03:5022',
AVAILABILITY_MODE =SYNCHRONOUS_COMMIT,
FAILOVER_MODE =EXTERNAL,
SEEDING_MODE =AUTOMATIC
)
```

Nachdem die Gruppe erfolgreich erstellt wurde, können die Replikate hinzugefügt werden. Hierfür gewähren wir der Gruppe die Erlaubnis zum Erstellen einer Datenbank. Dafür folgendes Skript ausführen:

```
ALTER AVAILABILITY GROUP [ubag] GRANT CREATE ANY DATABASE
GO
```

Mit folgendem Kommando kann nun eine Datenbank zu der Gruppe hinzugefügt werden:

```
ALTER AVAILABILITY GROUP [ubag] ADD DATABASE [UBAG_Sample];
```

Als letzten Schritt müssen wir jetzt noch Pacemaker Ressourcen erstellen, damit die Gruppe automatisch ein failover durchführen kann.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Erstellen der Pacemaker Cluster Ressourcen

Um nun die Pacemaker Cluster Ressourcen zu erstellen, führen wir auf unserer primären Node dieses Kommando aus:

```
sudo pcs resource create ag_cluster ocf:mssql:ag ag_name=ubag meta failure-
timeout=30s --master meta notify=true
```

Nun wählen wir eine unbenutzte IP-Adresse aus unserem Netzwerk und erstellen im Pacemaker eine virtuelle IP Ressource:

```
sudo pcs resource create virtualip ocf:heartbeat:IPaddr2 ip=172.31.6.225 cidr_
netmask=20
```

Jetzt müssen wir Colocation- und Ordnungsbeschränkungen für virtuelle IP-Ressourcen hinzufügen. Diese Einschränkung stellt sicher, dass die virtuelle IP auf dem aktiven Knoten online geschaltet wird, auf dem die Ressource ag_cluster online ist:

```
sudo pcs constraint colocation add virtualip ag_cluster-master INFINITY with-rsc-
role=Master
sudo pcs constraint order promote ag_cluster-master then start virtualip
```

Nun ist unsere Hochverfügbarkeitsgruppe erstellt und einsetzbar.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612