Mustafa Girgin 21290649

https://github.com/antelcha/sqlprojeler

Veritabanı Replikasyonu ve Dağıtık Yapı Kurulumu

SQL Server üzerinde iki farklı instance arasında veritabanı replikasyonu kurulumunu adım adım gerçekleştirdim. Buradaki amacım, bir Publisher (Yayıncı) ve bir Subscriber (Abone) arasında veri senkronizasyonunu sağlamaktı.

Docker Container'larının Hazırlanması İlk olarak, iki ayrı SQL Server instance'ını Docker üzerinde çalıştırmak için gerekli yapılandırmayı yaptım. Docker Compose dosyasında her iki sunucu için SQL Server Agent'ı aktif ettim ve gerekli port yönlendirmelerini tanımladım.
[photo]

Publisher Sunucusunun Yapılandırılması Publisher rolündeki ilk SQL Server instance'ında (sql_server_1) sırasıyla şu adımları uyguladım:
a) Öncelikle distributor kurulumunu gerçekleştirdim ve distribution database'ini oluşturdum:

Subscriber Sunucusunun Yapılandırılması İkinci SQL Server instance'ında (sql_server_2) subscription ayarlarını yaptım:
a) Önce subscriber veritabanını oluşturdum:
b) Push subscription tanımlamasını gerçekleştirdim:

Subscriber'da replikasyon kontrolü: Eklenen verilerin subscriber'a başarıyla replike olduğunu gözlemledim.

Bu testler, replikasyon yapılandırmasının beklendiği gibi çalıştığını gösterdi. Veriler publisher'dan subscriber'a sorunsuz şekilde aktarıldı.

Tüm bu adımları otomatize etmek için üç ayrı SQL script hazırladım:

1-publisher-setup.sql: Publisher kurulum adımları
2-subscriber-setup.sql: Subscriber kurulum adımları
3-test-replication.sql: Test senaryoları
Bu scriptler sayesinde replikasyon kurulumunu hızlı ve hatasız şekilde tekrarlayabiliyorum. Herhangi bir sorun yaşandığında SQL Server error log'larını kontrol ederek hatanın kaynağını kolayca tespit edebiliyorum.

```
-- 1. PUBLISHER SETUP (sql_server_1'de çalıştırılacak)
USE master;
GO
```

```
-- Mevcut distributor yapılandırmasını temizle
EXEC sp_dropdistributor @no_checks = 1, @ignore_distributor = 1;
GO

-- Replikasyon dağıtıcısını yapılandır
EXEC sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO

-- Distribution database'ini oluştur
USE master;
GO
IF NOT EXISTS (SELECT * FROM sys.databases WHERE name = 'distribution')
BEGIN
   -- Local distributor olarak ayarla
   EXEC sp_adddistributor
       @distributor = @@SERVERNAME,
       @password = 'Distr!bution2024';

   -- Distribution database'ini oluştur
   EXEC sp_adddistributiondb
       @database = 'distribution';
END
GO

-- Test veritabanını oluştur
USE master;
GO
IF EXISTS (SELECT * FROM sys.databases WHERE name = 'TestDB')
   DROP DATABASE TestDB;
GO
CREATE DATABASE TestDB;
GO

-- Veritabanını replikasyon için hazırla
USE TestDB;
GO
EXEC sp_replicationdboption
   @dbname = 'TestDB',
   @optname = 'publish',
   @value = 'true';
GO
```

```
-- Test tablosunu oluştur
CREATE TABLE Customers (
   CustomerID INT PRIMARY KEY IDENTITY(1,1),
   CustomerName NVARCHAR(100),
   CreatedDate DATETIME DEFAULT GETDATE()
);
GO

-- Yayıncı sunucuyu yapılandır
USE [TestDB]
GO
EXEC sp_addpublication
   @publication = 'CustomerPublication',
   @description = 'Publication for Customers table',
   @sync_method = 'concurrent',
   @retention = 0,
   @allow_push = 'true',
   @allow_pull = 'true',
   @allow_anonymous = 'true',
   @enabled_for_internet = 'false',
   @snapshot_in_defaultfolder = 'true';
GO

-- Yayın için makale ekle
USE [TestDB]
GO
EXEC sp_addarticle
   @publication = 'CustomerPublication',
   @article = 'Customers',
   @source_owner = 'dbo',
   @source_object = 'Customers',
   @type = 'logbased',
   @description = 'Customer table article',
   @creation_script = NULL,
   @pre_creation_cmd = 'drop',
   @schema_option = 0x000000000803509D,
   @identityrangemanagementoption = 'manual',
   @destination_table = 'Customers',
   @destination_owner = 'dbo';
GO

-- Snapshot Agent'ı başlat
EXEC sp_startpublication_snapshot @publication = 'CustomerPublication';
GO
```

```
-- 2. SUBSCRIBER SETUP (sql_server_2'de çalıştırılacak)
USE master;
GO

-- Eğer varsa subscriber database'i sil
IF EXISTS (SELECT * FROM sys.databases WHERE name = 'TestDB')
   DROP DATABASE TestDB;
GO

-- Subscriber database'i oluştur
CREATE DATABASE TestDB;
GO

-- Subscription oluştur
USE TestDB;
GO

-- Push subscription oluştur
EXEC sp_addsubscription
   @publication = 'CustomerPublication',
   @subscriber = 'sql_server_2',
   @destination_db = 'TestDB',
   @subscription_type = 'Push',
   @sync_type = 'automatic',
   @article = 'all',
   @update_mode = 'read only';
GO

-- Push subscription agent'ı oluştur
EXEC sp_addpushsubscription_agent
   @publication = 'CustomerPublication',
   @subscriber = 'sql_server_2',
   @subscriber_db = 'TestDB',
   @subscriber_security_mode = 0,
   @subscriber_login = 'sa',
   @subscriber_password = 'yourStrong(!)Password';
GO
```

```
-- 3. TEST REPLICATION
```

```sql
-- Publisher'da çalıştırılacak (sql_server_1):
USE TestDB;
GO

-- Test verisi ekle
INSERT INTO Customers (CustomerName)
VALUES
   ('Test Customer 1'),
   ('Test Customer 2'),
   ('Test Customer 3');
GO

-- Eklenen verileri kontrol et
SELECT * FROM Customers;
GO

-- Subscriber'da çalıştırılacak (sql_server_2):
USE TestDB;
GO

-- Replike olan verileri kontrol et
SELECT * FROM Customers;
GO

-- Not: Veriler subscriber'a replike olduysa,
-- publisher'da eklenen verilerin aynısını subscriber'da görmelisiniz.
```