

CMS 버전 3.0의 데이터베이스 클러스터를 구성하는 방법

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[구성](#)

[네트워크 다이어그램](#)

[구성](#)

[다음을 확인합니다.](#)

소개

이 문서에서는 단계별로 Cisco CMS(Meeting Server) 데이터베이스(DB) 클러스터 컨피그레이션에 대해 설명합니다.

기고자: Amadeus Ubaldo 및 Octavio Miralrio, Cisco TAC 엔지니어

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- CMS 일반 컨피그레이션
- SSH(Secure Shell) 클라이언트
- SFTP(Secure File Transfer Protocol) 클라이언트

사용되는 구성 요소

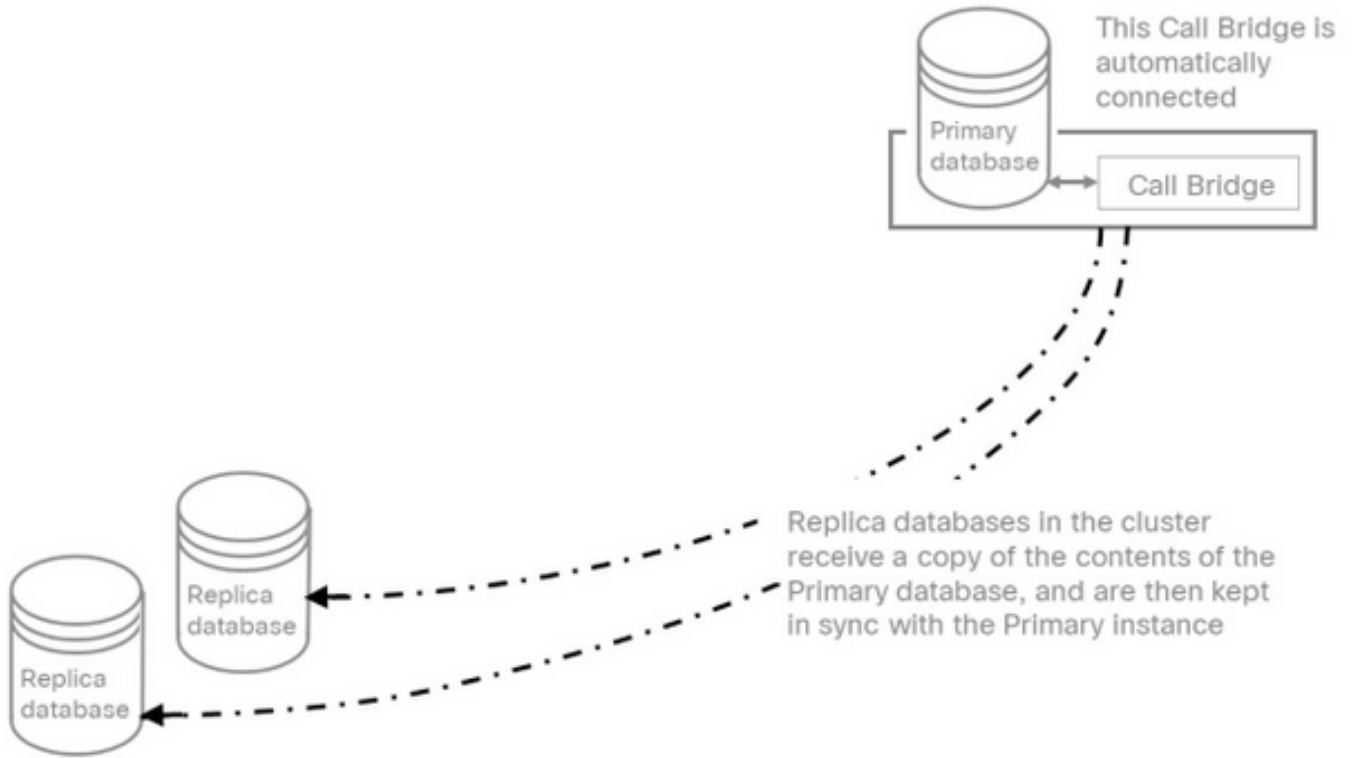
이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- CMS 버전 3.0
- Putty SSH 클라이언트
- Windows Secure Copy(winSCP) 클라이언트

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

구성

네트워크 다이어그램



구성

1단계. 데이터베이스 클러스터의 기본 노드가 될 CMS 하나를 선택합니다.

참고: CMS 데이터베이스 클러스터에는 홀수 서버와 최대 5개의 서버가 필요합니다. 자세한 [CMS 컨피그레이션 가이드](#)는 설명서를 참조하십시오.

2단계. 기본 역할에 대해 선택한 CMS에서 SSH 클라이언트를 사용하여 MMP(Mainboard Management Processor) 인터페이스를 엽니다.

3단계. 클라이언트 역할에 대한 CSR(Certificate Signing Request)을 생성합니다. 다음 형식을 사용합니다.

참고: 클라이언트 CSR의 CN(Common Name)은 postgres여야 합니다.

pki csr

```
172.16.85.108 - PuTTY
cms01>
cms01> pki csr DBClient CN:postgres
.....
.....
Created key file DBClient.key and CSR DBClient.csr
CSR file DBClient.csr ready for download via SFTP
```

4단계. 서버 역할에 대한 CSR을 생성합니다. 다음 형식을 사용합니다.

참고:SAN(Subject Alternative Name)과 동일한 데이터베이스 클러스터의 다른 서버 부분에서 FQDN(Fully Qualified Domain Names)을 추가합니다.

pki csr

```
cms01> c-cmspod3.lab subjectAltName:cms02.mxc-cmspod3.lab,cms03.mxc-cmspod3.lab
.....
.....
Created key file DBServer.key and CSR DBServer.csr
CSR file DBServer.csr ready for download via SFTP
cms01> █
```

5단계. 이전 명령에서 사용한 이름으로 생성된 파일이 4개 있는지 확인합니다.

DBServer.csr

DBServer.key

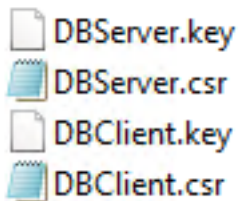
DBClient.csr

DBClient.key

6단계. DB 클러스터의 나머지 CMS 서버와 함께 3단계와 4단계를 수행합니다.

참고:첫 번째 서버에서 생성된 동일한 클라이언트 및 서버 CSR 및 키 쌍을 클러스터의 나머지 피어에 사용할 수 있습니다.이렇게 하려면 **.key** 파일 및 SFTP 소프트웨어를 사용하여 나머지 CMS 서버에 이미 서명된 인증서를 복사합니다.

7단계. CSR 파일을 수집하기 위해 기본 설정의 SFTP 소프트웨어를 사용하여 각 서버에 액세스합니다.



8단계. 각 서버의 CSR 파일을 인증서 서명에 사용할 CA(Certificate Authority)에 제공합니다.

9단계. 인증서가 이미 서명되면 SFTP 클라이언트를 사용하여 서명된 인증서를 각 CMS 서버에 각각 업로드합니다.

10단계. CLI를 통해 CMS 서버에 액세스하고 인증서 및 키 파일을 데이터베이스 클러스터 서비스에 할당합니다. 다음 구문을 사용합니다.

```
database cluster certs
```

11단계. 나머지 CMS 서버에 대해 10단계를 반복하여 인증서를 각각 할당합니다.

12단계. CLI를 통해 DB 클러스터의 마스터 노드로 지정된 CMS에 액세스한 다음 DB 서비스에 사용하도록 지정된 인터페이스를 선택합니다. 이 예에서는 인터페이스 a이고 다음 명령을 실행합니다.

```
database cluster localnode a
```

13단계. CMS DB 클러스터에 포함될 예정인 나머지 CMS 서버에 대해 12단계를 반복합니다.

14단계. CLI를 통해 마스터 DB 역할이 있는 CMS에 액세스하여 데이터베이스 서비스를 초기화하려면 다음 명령을 실행합니다.

```
database cluster initialize
```

참고: 확인은 대/소문자를 구분하므로 대문자 Y를 대문자로 입력해야 합니다. 이 명령은 다음 노드가 아닌 데이터베이스 기본 에서만 실행해야 합니다.

15단계. 다음 경고가 표시됩니다. Y를 선택합니다.

```
database cluster initialize WARNING!!! Are you sure you wish to initialize this node as a new database cluster?(Y/n) The contents of this node's database will become the primary version of the database in the new cluster. Initialization started...
```

16단계. 초기화 상태를 확인하고 명령 데이터베이스 클러스터 상태를 실행합니다.

```
Status: initializing Nodes: 172.16.85.108 (me): Connected primary Interface: a
```

참고: 상태는 데이터베이스 서비스 초기화 프로세스가 아직 완료되지 않은 경우 초기화를 표시합니다.

17단계. 데이터베이스 클러스터 상태가 활성화될 때까지 명령을 실행하여 다음 출력에 표시된 대로 초기화 프로세스가 성공했음을 의미합니다.

```
Status: Enabled Nodes: 172.16.85.108 (me) : Connected primary Interface: a
```

18단계. CLI를 통해 데이터베이스 클러스터에 속하는 다음 CMS 피어에 액세스합니다.

19단계. 명령 데이터베이스 클러스터 조인 <primary_database_ip_address>를 실행합니다.

20단계. Y를 선택합니다.

```
cms> database cluster join 172.16.85.108 WARNING? ! ! Are you sure you wish to attach this node to an existing database cluster? (Y/n) The contents of this node's database will be destroyed! Attachment started...
```

21단계. 나머지 서버에 대해 18, 19, 20단계에 설명된 절차를 반복합니다.

다음을 확인합니다.

CLI에서 직접 데이터베이스 클러스터의 상태를 확인할 수 있습니다. 명령 데이터베이스 클러스터 상태를 실행합니다. 다음 출력은 클러스터 프로세스가 아직 완료되지 않았음을 보여줍니다.

```
cms> database cluster status Status: Attaching Nodes: 172.16.85.108: Connected Primary  
172.16.85.117 (me): Connected Replica 172.16.85.118: Connected Replica Interface: a
```

데이터베이스 클러스터 조인 프로세스가 성공적으로 완료되면 상태가 사용으로 **설정됨**으로 표시되어야 합니다.

```
cms> database cluster status Status: Enabled Nodes: 172.16.85.108: Connected Primary  
172.16.85.117 (me): Connected Replica 172.16.85.118: Connected Replica Node in use:  
172.16.85.108 Interface: a Certificates Server Key: DBServer.key Server Certificate:  
DBServer.cer Client Key: DBClient.key Client Certificate: DBClient.cer CA Certificate:  
CA_root.cer Last command: 'database cluster join 172.16.85.108' (Success)
```