Configure Meeting Server (CMS) Call Bridge Database Cluster

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Configure

- Part 1. Certificate Creation
- Part 2. Call Bridge Configuration
- Network Diagram

Verify

Troubleshoot

Related Information

Introduction

This document describes the steps to configure the database (DB) clustering on Cisco Meeting Server (CMS) or Acano call bridges (CB).

Prerequisites

Requirements

- Cisco recommends that you have at least 3 CMS nodes to be able to create a viable DB cluster

Note: It is recommended to have an odd number of DB cluster nodes as it is important for the primary selection and the active failover mechanism. Another reason for this is that the primary DB node would be the node that has connections to the most of the DB in the cluster. Only 3 nodes can be joined to a database cluster and any further servers can be added to the cluster using **database cluster connect** command.

- Port 5432 opened on firewall

Note: The DB cluster Primary node listens on port 5432 for connections from the Replica nodes, so if there's a firewall (FW) between the nodes, ensure that this port is opened.

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the

devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Configure

There are two types of certificates for the DB clustering:

1. Client: The client certificate, as the name sugest, is used by the DB clients to connect to the DB server (primary). This certificate must contain the string, postgres, in its Common Name (CN) field.

2. Server: The server certificate, as the name sugest, is used by the DB server to connect to the postgres DB.

Part 1. Certificate Creation

1. Connect with a Secure Shell (SSH) with the admin credentials to the server MMP.

2. Generate Certificate Signing Request (CSR):

a. For the databasecluster client certificate:

pki csr <key/cert basename> CN:postgres

For example: **pki csr databasecluster_client CN:postgres**

b. For the databasecluster server certificate:

pki csr <key/cert basename> CN:<domainname>

For example: **pki csr databasecluster_server CN:vngtpres.aca**

3. Send the CSRs to your Cettificate autority (CA) to have them signed. Ensure that the CA provides you with the Root CA (and any intermediate CA) certificates.

4. Upload the signed certificates, Root CA (and any intermediate CA) certificates onto all the DB nodes using an Secure File Transfer Protocol (SFTP) client (for example WinSCP).

Note: The CN for Part A must be postgres and Part B can be the domain name of the call bridge, no Subject Alternate Name (SAN) entries are required.

Part 2. Call Bridge Configuration

On the CB that runs the primary DB, follow these steps:

1. To select the interface to use, enter the command:

database cluster localnode a

This enables interface “a” to be used for the DB clustering.

2. Define the client, server and root ca certificates as well as the private keys to be used by the DB cluster with these commands:

database cluster certs <client_key> <client_crt> <ca_crt>

database cluster certs <server_key> <server_crt> <client_key> <client_crt> <ca_crt>

Note: The same client and server certificates can be used on other CB nodes to be clustered when you copy the private keys and certificates across to the other nodes. This is possible because the certificates contain no SAN tying them to a specific call bridge. However, it is recommend to have individual certificates for each DB node.

---

3. Initialize this DB on the local CB as the primary for this DB cluster:

database cluster initialize

4. On the CallBridges that would be part of the clustered DB and become the DB replica run this command after you complete Steps 1 and 2 for Part 2:

database cluster join <Primary CB IP address>

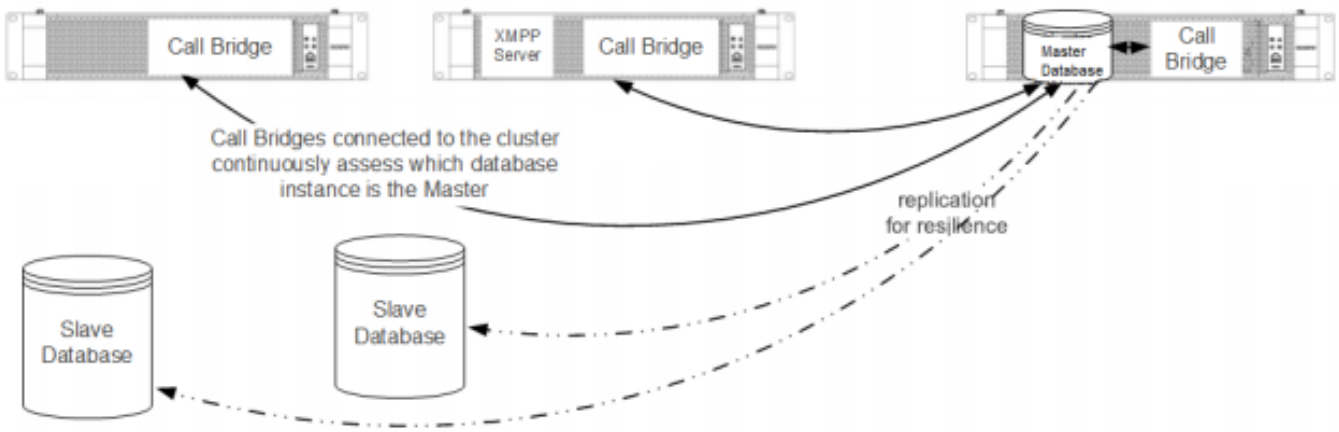
For example: **database cluster join <10.48.36.61>**

```
CMS> database cluster join
WARNING!!!
Are you sure you wish to join this node to an existing database cluster? (Y/n)
 The contents of this node's database will be destroyed!
 The callbridge and web administration will restart at the end of this procedure.
```

This initiates the DB synchronization and copy the DB from the primary peer.

Note: The **local DB** that existed before the **database cluster join** command is going to be overwritten, and therefore its content is going to be deleted.

Network Diagram



Verify

Use this section to confirm that your configuration works properly.

To check the clustered DB status, run this command on any of the nodes in the DB cluster:

database cluster status

The output is similar to:

```
  Status                   : Enabled
  Nodes:
      10.48.36.61          : Connected Primary
      10.48.36.118         : Connected Replica ( In Sync )
      10.48.36.182 (me)    : Connected Replica ( In Sync )
  Node in use              : 10.48.36.61

      Interface            : a

      Certificates
      Server Key           : dbclusterserver.key
      Server Certificate   : dbclusterserver.cer
      Client Key           : dbclusterclient.key
      Client Certificate   : dbclusterclient.cer
      CA Certificate       : vngtpRootca.cer
  Last command             : 'database cluster join 10.48.36.61' (Success)
```

Troubleshoot

This section provides information you can use in order to troubleshoot your configuration.

Use this command, on the CLI, to view the current logs related to the DB clustering:

syslog follow

Log outputs for the DB usually contain the postgres string, examples as follows:

<#root>

```
Mar 30 12:39:04 local0.warning ClusterCore1

postgres

[20882]:  [2-7] #011SQL statement "INSERT INTO domains(domain_id, domain_name, tenant_id, target, prior
Mar 30 12:39:04 local0.warning ClusterCore1

postgres

[20882]:  [2-8] #011PL/pgSQL function create_or_update_matching_domain(boolean,uuid,text,boolean,uuid,i
Mar 30 12:39:04 local0.warning ClusterCore1

postgres

[20882]:  [2-9] #011SQL statement "SELECT * FROM create_or_update_matching_domain(TRUE, inp_domain_id,
Mar 30 12:39:04 local0.warning ClusterCore1

postgres

[20882]:  [2-10] #011PL/pgSQL function create_matching_domain(uuid,text,uuid,integer,integer,text) line
```

Here are some typical DB issues and solutions:

Problem: DB schema error on a non-primary peer

```
ERROR                   : Couldn't upgrade the schema
Status                  : Error

Nodes:
    10.48.54.75         : Connected Primary
    10.48.54.76         : Connected Replica ( In Sync )
    10.48.54.119 (me)   : Connected Replica ( In Sync )
Node in use             : 10.48.54.75

Interface               : a

Certificates
  Server Key            : dbclusterServer.key
  Server Certificate    : dbserver.cer
  Client Key            : dbclusterClient.key
  Client Certificate    : dbclient.cer
  CA Certificate        : Root.cer

Last command            : 'database cluster upgrade_schema' (Failed)
```

Solution:

1. First, run this command to clear the error:

database cluster clear error

2. Followed by this command to upgrade the DB schema:

database cluster upgrade_schema

3. Then check to the status of the DB clustering with:

database cluster status

The logs show output similar to this:

<#root>

```
Mar 30 11:22:45 user.notice acanosrv05 schema_builder:  Upgrading schema with connect line 'connect_tim
```

client.key

```
 sslrootcert=/srv/pgsql/ca.crt '
```

```
Mar 30 11:22:45 user.notice acanosrv05 schema_builder:  Using database name 'cluster'
Mar 30 11:22:45 user.notice acanosrv05 schema_builder:  schema build on database cluster complete
Mar 30 11:22:45 user.notice acanosrv05 schema_builder:  Using CiscoSSL 1.0.1u.4.13.322-fips  (caps 0x4F
Mar 30 11:22:45 user.notice acanosrv05 schema_builder:  Using 0x1000115F
Mar 30 11:22:45 user.notice acanosrv05 schema_builder:  INFO    : Waiting for database cluster to settl
Mar 30 11:22:45 user.notice acanosrv05 schema_builder:  INFO    : Database cluster settled
Mar 30 11:22:45 user.notice acanosrv05 schema_builder:  Schema upgrade complete
Mar 30 11:22:45 user.info acanosrv05 dbcluster_watcher:  Operation Complete
```

Problem: Peer nodes unable to connect to DB primary node

<#root>

```
Mar 31 10:16:59 user.info acanosrv02 sfpool:  Health check 10.48.54.119: error (up = 1): could not conn
```

Connection refused

```
|#011Is the server running on host "10.48.54.119" and accepting|#011TCP/IP connections on port
```

5432

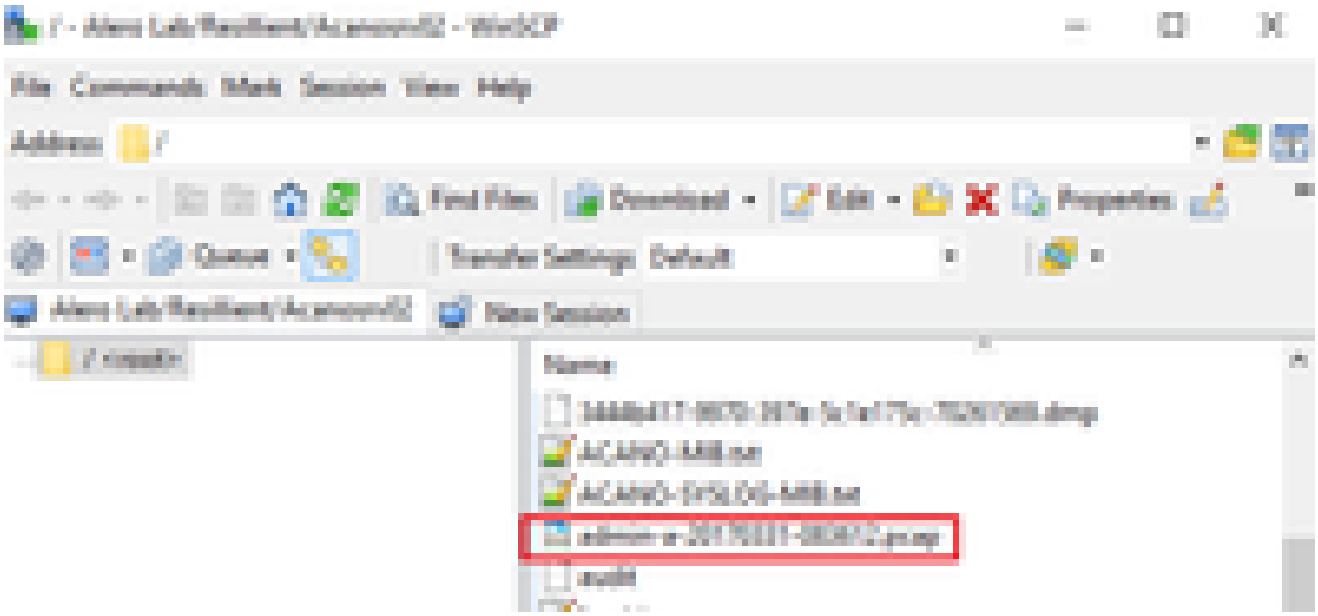
```
?|
```

Solution:

Use these steps to collect traces to troubleshoot the connection issues:

1. Run the command **pcap <interface>** on the non-primary (replica) node and after a few minutes, stop the capture with **Ctrl-C**.

2. Connect with a Secure File Transfer Protocol (SFTP) client to the server and download the **.pcap** file from the root directory:

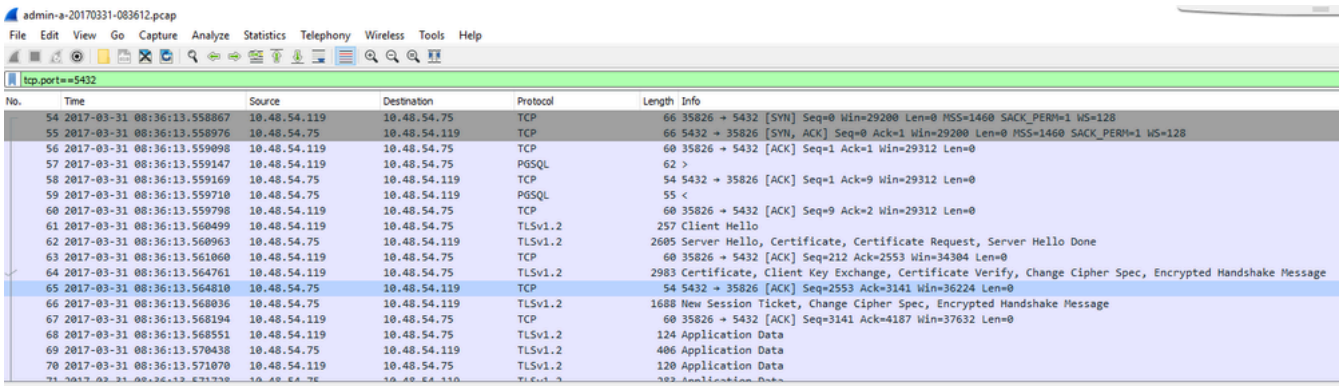


3. Open the capture file on Wireshark and filter on port 5432 with **tcp.port==5432** to check for traffic between the non-primary peer and the DB primary.

4. If there is no return traffic from the server then it is likely that a FW can be blocking the port between the two servers' logical location.

Here is a typical packet capture from a working connection betwen the client and server:

In this example the client IP is 10.48.54.119 and the server is 10.48.54.75.



Related Information

For more infromation how to troubleshoot issues, and other questions on Database clustering, refer to the FAQs in these links:

- When clustering database servers, why do I need to place them on different locations
- We have a database cluster, and I see a database error or warning in the log. What should I do
- One or more of the database servers are not connected or in "Sync" state. What should I do
- What do I do if there is no primary database
- How do I move the primary database
- Technical Support & Documentation - Cisco Systems