How to Move the PostgreSQL Database Between Inter-Cluster Peers in IM & Presence (IM&P)

Contents

Introduction

This document describes how to move the PostgreSQL database between inter-cluster peers in Instant Messaging (IM) and Presence (IM&P).

Contributed by Joel Burleigh and edited by Joseph Koglin, Cisco TAC Engineers

Prerequisites

Requirements

Cisco recommends you have an environment that meet these conditions.

- You have two IM&P clusters with version(s) 9.1.1
- Inter-Cluster peers were setup between IM&P cluster.
- PostgreSQL database was configured on one cluster with a single database instance associated with the IM&P Publisher

Components Used

The information in this document is based on these software version(s) and components:

- IM&P Version(s) 9.1.1
- PostgreSQL

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.

Solution

To eliminate any confusion these terms will be used to refer to the IM&P Clusters and the PostgreSQL database.

Terms

- **Cluster1** - the IM&P cluster where the external database was originally configured
- **Cluster2** - the IM&P cluster where the external database configuration will be moved to.
- **PostgreSQL** - for the external database host

Steps performed

> **Warning**: These steps should only be utilized if you are left with no other options. Before you proceed with these steps please discuss internally to ensure this is your best solution.

> **Note**: It should be noted that if Perssitent chat is setup correctly for Inter-Cluster peers. Each node on each cluster should have it own database instance on PostgreSQL. The only exception to this is if the version is 11.5 and above.

Step 1. First input the CLI command from the IM&P Publisher where your database is currently hosted (Cluster1).

```
run sql select * from tcaliases
```

Make note of the dynamically created conference ID and manually created alias associated to the local cluster.

An example of a dynamic conference ID is conference-2-StandAloneCluster2c2aa.jburleig.local. You can tell this is the primary conference ID as the primary is set to true and it has a value in the fkprocessnode field

An example of a chat node alias is pchat1.jburleig.local you can tell this because the primary is set to false but it has the same pkid value in the fkprocessnode column as primary conference ID does.

Example Output:

```
admin:run sql select * from tcaliases
pkid                                  tcalias
isprimary  fkprocessnode                        peerclusterid
====================================  ==================================================
=========  ==================================== ==============
50a4cf3b-0474-4723-ba50-4cd2cc1dd277  conference-2-StandAloneCluster2c2aa.jburleig.local
  t      2c2aa1f6-cc7a-470a-a0ba-c8a892db68ca  NULL
9eca651d-5a67-3116-a57b-
1eb2ab0911bd  pchat1.jburleig.local                                   f      2c2aa1f6-cc7a-470a-
a0ba-c8a892db68ca  NULL
838e900a-0d2f-4843-be00-ac0a6c803ab5  conference-2-StandAloneClustercbea5.jburleig.local
f      NULL                                  2202
```

Step 2. Create a backup of the current database (PostgreSQL).

> **Note**: This should be done by your database administrator in accordance with your organizations requirements.

Step 3. Next create a new database instance (PostgreSQL)

- The table created in command below is named *cluster2* and the user used to create it is tcuser.

Note: The Encoded method of the database could be different other then UTF8.

```
CREATE DATABASE cluster2 WITH OWNER tcuser ENCODING 'UTF8'
```

Step 4. You will need to add a new entry to allow access to the tcuser to the new database created in step 2.

If the new external database configuration on the IM&P cluster will be on a new IP subnet make sure to update the subnet in the entry you make in the **pg.hba.conf** file (PostgreSQL).

- Add the entry to the ***install_dir*/data/pg_hba.conf**

```
host    DBName      DBUsere     Subnet          password
host    cluster2    tcuser      10.10.1.0/24    password
```

Step 5. Next you will need to create a new external database on the IM&P cluster that the configuration will be moved to (Cluster2)

- Navigate to **Messaging > External Server Setup > External Databases**
  - Select the **Add New** button and configure the new external databse and use the database name and user created in the step 3

Step 6. Now disable persistent chat on the current IM&P that hosts the Persistent chat configuration and unassign the external database that is associated to the Persistent chat configuration (Cluster1)

- Navigate to **Messaging > Group Chat and Persistent Chat**
  - Uncheck the **Enable Persistent Chat** checkbox
  - Set the External Database to unassigned

Step 7. Next delete the external database configuration (Cluster1)

- Navigate to **Messaging > External Server Setup > External Databases**
  - Select the checkbox next to the configured PostgreSQL persistent chat database and select delete.

Step 8. Next delete the persistent chat custom alias configured on the current cluster (Cluster1)

- Navigate to **Messaging > Group Chat Server Alias Mapping**
  - Select the checkbox next to the configured Alias and select delete.

Step 9. Once the persistent chat and external database configuration has been completely removed (Cluster1) Restart the Cisco XCP Router (Cluster1**)**

Step 10. Next enable persistent chat on (Cluster2) and assign external database that was created in step 5.

- Navigate to **Messaging > Group Chat and Persistent Chat**
  - Check the Enable Persistent Chat checkbox
  - Set the External Database to the external database that was created in step 5.

Step 11. Verify the external database connectivity test shows ok on (Cluster2) after you enable persistent chat, only continue if it has all green check marks.

- Navigate to **Messaging > External Server Setup > External Databases** Verify that the external database troubleshooter shows green check marks.

- Select the configured External Database that was setup in step 5

Step 12. Create a custom alias on (Cluster2) Make sure to use the exact name of the one you deleted from the old cluster. You can find the name of the alias in the output from step 1.

- Navigate to **Messaging > Group Chat Server Alias Mapping > Add New**
- Add the exact same alias that use to be configured on Cluster1. You can find the alias name in the command output from Step1.

Step 13. Next Restart the XCP router (Cluster2)

Step 14. Once the Cisco XCP Router has restarted successfully (Cluster2) proceed and stop the Cisco Text Conferencing Manager (Cluster2)

Step 15. Perform a database restore with the PostgreSQL backup that was made in step 2. Make sure to restore the backup to the new database instance that was created in step 3 (PostgreSQL)

- Verify that the imported data is in the tc_rooms, tc_users, tc_messages and tc_msgarchive tables.

Step 16. Next restart the PostgreSQL service (PostgreSQL)

Step 17. Next start the text conferencing manager on cluster2.

- At this point if a user logins to Jabber they will be able to see any of the chat rooms that were created with the persistent chat Alias. Any chat rooms that are associated to the dynamically assigned chat room conference ID will not show as they are currently associated to the old cluster. You will fix this by running SQL update commands on the PostgreSQL server.

Step 18. From the PostgreSQL command line execute these commands to update the old conference ID to the new alias that was created in step 12. (PostgreSQL)

> **Note**: You will need to adapt these commands to contain your Cluster1 conference ID and the alias ID you have configured.

```
Updates for tc_rooms

update tc_rooms set room_jid = replace(room_jid, ‘conference-2-StandAloneCluster2c2aa.jburleig.local’, ‘pchat1.jburleig.local’);

Updates for tc_users

update tc_users set room_jid = replace(room_jid, ‘conference-2-StandAloneCluster2c2aa.jburleig.local’, ‘pchat1.jburleig.local’);
update tc_users set nick_jid = replace(nick_jid, ‘conference-2-StandAloneCluster2c2aa.jburleig.local’, ‘pchat1.jburleig.local’);
update tc_users set initiator_jid = replace(initiator_jid, ‘conference-2-StandAloneCluster2c2aa.jburleig.local’, ‘pchat1.jburleig.local’);

Updates for tc_messages

update tc_messages set room_jid = replace(room_jid, ‘conference-2-StandAloneCluster2c2aa.jburleig.local’, ‘pchat1.jburleig.local’);
update tc_messages set msg = replace(msg, ‘conference-2-StandAloneCluster2c2aa.jburleig.local’,
‘pchat1.jburleig.local’);
```

```
Updates for tc_msgarchive
```

```
update tc_msgarchive set to_jid = replace(to_jid, ‘conference-2-
StandAloneCluster2c2aa.jburleig.local’, ‘pchat1.jburleig.local’);
update tc_ msgarchive set nick_jid = replace(nick_jid, ‘conference-2-
StandAloneCluster2c2aa.jburleig.local’, ‘pchat1.jburleig.local’);
update tc_ msgarchive set message_string = replace(message_string, ‘conference-2-
StandAloneCluster2c2aa.jburleig.local’, ‘pchat1.jburleig.local’);
```

Step 19. Next restart the PostgreSQL service (PostgreSQL)

Step 20. Next restart the text conferencing manager (Cluster2)

Step 21. At this point Jabber clients should be able to login to IM&P and retrieve all rooms in the All Rooms tab.