Telephony Capture Service

Quick Start Guide

Version 1.0.0



**Document Modification History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Editor** | **Date** | **Modification** |
| 1.0.0 | R Monk | 2017-01-31 | Original Release |

Table of Contents

1 Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Document Location 4

1.4 Document Status 4

1.5 Acronyms Definitions 4

1.6 Reference and Related Documents 4

1.7 Open Issues 4

2 Basic TCS Usage 5

3 Database Management 6

4 Version Management 7

# Introduction

## Purpose

ToDo

## Scope

ToDo

## Document Location

This document is found in the ‘docs’ folder of the TCS GitHub repository:

<https://github.com/ccbcadmin/telephony-capture-service.git>

## Document Status

This document must be kept current and released concurrently with each software release.

## Acronyms Definitions

The reader is referred to the [TCS Software Requirements Document](TCS Software Requirements Document.docx)

## Reference and Related Documents

* [TCS Software Requirements Document](TCS Software Requirements Document.docx)
* [TCS Developer Manual](TCS Developer Manual.docx)
* [TCS User Manual](TCS User Manual.docx)

## Open Issues

# Basic TCS Usage

1. Start with a clean sheet as follows:
   * $ rm –rf ~/tcs (remove all existing software)
   * User Kitematic to remove all containers
   * Remove all existing images as follows: $ docker rmi $(docker images –q)
   * Remove all existing volumes: $ docker volume rm $(docker volume ls –q)
2. Carry out the actions – start with version v0.27:
   * $ tcsproj v0.27
   * $ tcs
3. Confirm 2 ways that the 6 containers are running (barman, pg1, rabbitmq, database-interface, pbx-interface, and tms-interface):
   * Using Kitematic
   * $ docker ps
4. Right now nothing is actually happening, but you can get an input of SMDR traffic going with the following command:

$ pbx-simulator (one message per second).

* Confirm 7 Running containers
* There are 2 useful observations you can make at this point:
  + First, the tms-simulator is not yet running, which means that the queue PROD\_TMS\_QUEUE should be slowing growing in size.  You can check this on a browser with the following url: localhost:15672.  Select guest/guest and then select the menu Queues.  Look for PROD\_TMS\_QUEUE - it should be non-empty.
  + Execute the following commands:
    - $ psql1   
      This will open an SQL terminal session to pg1.
    - prod=# select count(\*) from smdr;   
      Execute this several times.  The total record count should be increasing.
    - \q (exits out of psql)
* Now start the tms-simulator with the following command:

$ tms-simulator

Check that there are now 8 running containers.  Use the RabbitMQ interface to check that indeed the number of PROD\_TM\_QUEUE messages is now zero.

# Database Management

This section simply asks you to try out the various database management utilities (list-backups, pg-switch, pg-rollback-recovery, pg-standby-recovery).

* $ list-backups
* $ pg-rollback-recovery backup\_id [pitr\_timestamp]  
  Note that  pitr\_timestamp is optional.  You need to run $ list-backups to get the backup\_id that you want.
* $ pg-offline-recovery backup\_id [pitr\_timestamp]  
  The parameters are the same as for pg-rollback-recovery, but this time the recovery is the pg2 container. Note that after executing this command, a new container, pg2, will exist and be running. You can connect to pg2 using port 5433.

# Version Management

* Play with version v0.27, v0.28, and v0.29 (I have specifically crafted these for this demo). Note that as you switch to a version for the first time, the software will need to download images from Docker Hub, hence at first startup there will be a delay (but note: if you switch back to the same version later, then the previously download images will still be available if you have not deleted them and hence the delay is minimal the second time).
* Now let us assume that a developer has been working on a feature request to be added to v0.28, but en route has left a goofy message (literally) in the console at the startup of **prod-pbx-interface** (you can see it in the Kitematic console display). Do this (assuming you are currently running v0.27).

$ tcs down

$ tcsproj v0.28

$ tcs

* Look for that goofy message.
* So, let us say that a developer fixes the problem and it is fixed in v0.29 and it is now available. Execute the following:

$ tcs down

$ tcsproj v0.29

$ tcs

* After this completes executing, you should see that the prod-pbx-interface ‘goofy message’ has been removed.
* Then just for fun, go back to v0.27.