



Managing Connected Analytics for Mobility (Wi-Fi)

CAM administrators perform many tasks that maintain the health and performance of the CAM BI platform and databases. Using the BI platform Administration Console, administrators can manage users and user roles, database connections, and database query schedules. Additional Administration functions include backing up the CAM BI platform and database and, if needed, restoring the BI platform and database.

Administrative functions and tasks are covered in the following topics:

- [Backing Up and Restoring CAM, page 6-1](#)
- [Starting and Stopping CAM Servers, page 6-4](#)

Please refer to [Managing Prime Analytics](#) in the *Cisco Connected Streaming Analytics 1.1 User Guide* for more management topics.

Backing Up and Restoring CAM

Backing up and restoring CAM involves two areas—Data Warehouse and BI Platform. The Data Mediation server continuously processes raw events and pushes the results to the Data Warehouse as soon as possible. It retains raw records for only the last five days. Thus, backing up the Data Mediation server is only useful if backups keep happening before processed data is pushed out to the Data Warehouse.

Backing Up the Data Warehouse

The Data Warehouse uses the PostgreSQL 9.3 database server. Therefore, you use the tools provided as part of PostgreSQL to backup the Data Warehouse. For detailed information on the options and tools for backup, refer to the PostgreSQL documentation at:

<http://www.postgresql.org/docs/9.3/static/backup.html>.

The following sections show simple examples with sample scripts for backup and restore using three different strategies.

Strategy 1: Use pg_dump and pg_restore

pg_dump and pg_restore commands offer a simple and straightforward way to backup a database. They can be executed as the primea user and do not require you to stop the database. The one major drawback of using dump and restore is that backups are not incremental. This means that as database size grows the time taken to dump and subsequently restore continues increasing.

```
%cat backup.sh
#!/bin/bash

test $# -lt 1 && echo "Please specify a backup directory" &&
exit 1

test ! -d $1 && echo "$1 directory does not exist!!!" &&
exit 1

touch $1/a > /dev/null
test $? -ne 0 &&
echo "cannot create file in $1. Please check permissions!!!" &&
exit 1

rm -f $1/a

site_env=/etc/sysconfig/primeanalytics/primea
test -e $site_env && source $site_env ||
(echo "no site env $site_env"; exit 1)

PATH=$PGHOME/bin:$PATH

pg_dumpall -r > $1/roles.sql

for db in $(psql -d postgres -U primea -A -t -q -c "select datname from
pg_catalog.pg_database where datistemplate is false")
do
    bkup_dir=${1}/${db}_backup
    echo "pg_dump -j 4 -F d -f $bkup_dir $db"
    pg_dump -j 4 -F d -f $bkup_dir $db
done

%cat restore.sh
#!/bin/bash

test $# -lt 1 &&
echo "Please specify directory to restore from" &&
exit 1

test ! -d $1 && echo "$1 directory does not exist!!!" &&
exit 1

site_env=/etc/sysconfig/primeanalytics/primea
test -e $site_env && source $site_env ||
(echo "no site env $site_env"; exit 1)

PATH=$PGHOME/bin:$PATH

if [ $(ls -A $1 | egrep "_backup$" | wc -l) -eq 0 ]; then
    echo "nothing to restore"
    echo "restore looks for \"*_backup\" directories"
    exit 0
fi

psql -q -d postgres -c "select 1" 2>/dev/null
test $? -ne 0 && echo "could not connect to database" &&
exit 1
```

```

test -f $1/roles.sql &&
psql -d postgres -f $1/roles.sql
for path in $(ls -ld $1/*_backup)
do
    base_dir=$(basename $path)
    db_name=${base_dir}_backup}
    db_exists=$(psql -d postgres -U primea -A -t -q -c "select count(1) from
pg_catalog.pg_database where datname = '$db_name'")

    if [ "$db_exists" -eq 0 ]; then
        STOP_ON_ERROR=1 psql -d postgres -U primea -c "create database $db_name"
        test $? -ne 0 && echo "failed to create database $db_name" &&
        exit 1
    fi
    test $? -eq 0 && echo "pg_restore -j 3 -F d $path -d $db_name" &&
    pg_restore -j 3 -F d $path -d $db_name

done

```

Strategy 2: Continuous Archiving

Continuous archiving is a better strategy for larger databases. Continuous archiving combines a file system level backup (base backup) of the PGDATA directory with backup of WAL files to offer a more incremental approach to backing up the Data Warehouse. It also gives you the ability to archive files on a backup server. To achieve this, the database has to be configured to archive WAL files before the first base backup of the database is taken.

Add the following to the PGDATA/postgresql.conf:

```

echo ""
wal_level = archive
archive_mode = on
archive_command = '/usr/bin/rsync -a --ignore-existing %p /archive/%f'
#limit how old unarchived data can be to 10 mins
archive_timeout = 600
"" > $PGDATA/postgresql.conf

#service postgresql restart #for changes to take effect

```

Now you can take base backups. A simple script is provided below as an example. Regular base backups can be performed by adding the script to:

```

%cat base_backup.sh
#!/bin/bash
site_env=/etc/sysconfig/primeanalytics/primea
test -e $site_env && source $site_env || (echo "no site env $site_env"; exit 1)

backup_dir=/backup/data
test ! -d $backup_dir && echo "no such directory $backup_dir !!!" && exit 1

touch a $backup_dir
test $? -ne 0 && echo "$backup_dir not writeable by user $(whoami)" && exit 1

export PGDATA PGPORT
export PATH=$PGHOME/bin:$PATH
psql -d postgres -c "select pg_start_backup('backup');"
rsync -a --progress $PGDATA/* $backup_dir
psql -d postgres -c "select pg_stop_backup();"

```

**Note**

Backups are performed as user primea. Backup and archive directories should be accessible and writable by user primea.

With WAL archiving, periodic cleanup of the archive directory specified in the `archive_command` in `postgresql.conf` is required to reclaim space used up by WAL files archived before a base backup. This can be done by using the `pg_archivecleanup` tool that comes with PostgreSQL. A typical invocation could be:

```
%pg_archivecleanup /path/to/archive_dir 000000010000000000000010.00000020.backup
```

In the example, `000000010000000000000010.00000020.backup` is the name of the file generated by the `pg_stop_backup()` function. The contents of this file show the start/stop WAL location, checkpoint location and start/stop time of the backup. The latest such file generated by `pg_stop_backup()` is located in the `$PGDATA/pg_xlog` directory.

Recovering from such a backup is discussed in detail in the section related to Continuous Archiving in the PostgreSQL 9.3 manual.

Strategy 3: Hot Standby

Setting up a hot standby not only can serve as a backup, but it can also be used for running read queries. See Chapter 4 “Installing CAM in a High Availability (HA) Configuration” in *Cisco Connected Analytics for Mobility (Wi-Fi) 1.1 Quick Start Guide*.

Backing up BI Platform

Pentaho stores meta data both on the file system, as well as in the database. For CAM, the default BI Platform is configured to store meta data on the local PostgreSQL data base. Thus, backing up the database automatically will backup the Pentaho data persisted in the database. But for a complete backup of Pentaho, the solutions folder also needs to be backed up. This is a simple file system level backup and can be performed by any of the standard Linux tools such as `tar`, `rsync` or `cp`. One way of doing this is shown in the following example:

```
# su - bipuser
% rsync -az $PA_PENTAHO_SOLUTIONS /pentaho_backups/backup_$(date +%F_%T)
```

Starting and Stopping CAM Servers

[Table 6-1](#) provides commands that you can use to start and stop the CAM BI platform and TruCQ engine. You must run these commands as the root user.

You must start the services in the following order: TruCQ engine, handlers, Data Warehouse server, and BI platform. Most of these services will automatically start on reboot, but the handlers must be started before the Data Warehouse server and BI platform will start.

You can start and stop the BI platform using the service `biplatform` command as the root. Alternatively, you can start and stop the BI platform using `./start-pentaho.sh` and `./stop-pentaho.sh` as the root.

Table 6-1 CAM Server Commands

Server	Action	Command
Data Mediation	Stop the TruCQ engine	service trucq stop
Data Mediation	Start the TruCQ engine	service trucq start
Data Mediation	Start the Handlers	
Data Mediation	Stop the Handlers	
Data Warehouse	Stop the Data Warehouse server	service postgresql stop
Data Warehouse	Start the Data Warehouse server	service postgresql start
Data Warehouse	Stop the BI platform	service biplatform stop
Data Warehouse	Start the BI platform	service biplatform start

Once the Data Mediation server is restarted, complete the following steps as root user:

Step 1 Verify that TruCQ is running. On server restart, TruCQ will be started automatically.

```
#service trucq status
```



Note This should show something like “trucq (pid) is running”.

Step 2 Start the data mover. For non-HA install:

```
#/opt/primeanalytics/mover/start.sh
```

For HA install:

```
#/opt/primeanalytics/mover/start_ha.sh
```

Step 3 Start the SNMP and AAA handlers:

```
#pushd /opt/primeanalytics/handlers/wifi
```

```
#!/build-local.sh
```

```
#!/start.sh
```

```
#popd
```

In the case of a Data Warehouse restart, both BI Platform and Postgresql will be started automatically. Verify they are started:

Step 1 Verify postgresql service is running:

```
# service postgresql status
```

Step 2 Verify biplatform is running:

```
# service biplatform status
```