Moving an Existing Postgres Database into an EDB Ark™ Cluster

Version 2.0

September 13, 2016

Moving an Existing Postgres Database into an EDB Ark Cluster
by EnterpriseDB® Corporation

EnterpriseDB Corporation, 34 Crosby Drive, Suite 100, Bedford, MA 01730, USA
T +1 781 357 3390 **F** +1 978 589 5701 **E** info@enterprisedb.com **www**.enterprisedb.com

Table of Contents

1 Introduction

EDB Ark automatically provisions PostgreSQL or EDB Postgres Advanced Server (Advanced Server) databases in single instances, high-availability clusters, or application development sandboxes in an OpenStack environment.

This tutorial describes recreating the content of an existing Postgres database into an equivalent EDB Ark cluster. While the examples demonstrate moving an Advanced Server database that resides on a Linux host, the same process will work with a PostgreSQL database or a Windows host. Additional steps may be required if the server versions differ. This tutorial assumes:

- You are a registered EDB Ark user.
- You have an existing PostgreSQL or Advanced Server database.
- You have created an EDB Ark cluster that is running the same version of Postgres as the source database.

In this tutorial, the term *Postgres* refers to EDB Postgres Advanced Server or PostgreSQL database.

EDB Ark uses public-key authentication; for detailed information about public-key authentication, please see the *EDB Ark Getting Started Guide*, available through the link on your EDB Ark dashboard.

> By default, only port `9999` on the master server node of an EDB Ark cluster is open for client connections. Before connecting with `ssh` or `scp`, an OpenStack Administrative user must first modify the security group for the EDB Ark cluster, opening port `22` for connections.

1.1 Typographical Conventions Used in this Guide

Certain typographical conventions are used in this manual to clarify the meaning and usage of various commands, statements, programs, examples, etc. This section provides a summary of these conventions.

In the following descriptions a *term* refers to any word or group of words that are language keywords, user-supplied values, literals, etc. A term's exact meaning depends upon the context in which it is used.

- *Italic font* introduces a new term, typically, in the sentence that defines it for the first time.
- `Fixed-width (mono-spaced) font` is used for terms that must be given literally such as SQL commands, specific table and column names used in the examples, programming language keywords, etc. For example, `SELECT * FROM emp;`
- *`Italic fixed-width font`* is used for terms for which the user must substitute values in actual usage. For example, `DELETE FROM` *`table_name`*;
- A vertical pipe | denotes a choice between the terms on either side of the pipe. A vertical pipe is used to separate two or more alternative terms within square brackets (optional choices) or braces (one mandatory choice).
- Square brackets [ ] denote that one or none of the enclosed terms may be substituted. For example, `[ a | b ]` means choose one of "`a`" or "`b`" or neither of the two.
- Braces {} denote that exactly one of the enclosed alternatives must be specified. For example, `{ a | b }` means exactly one of "`a`" or "`b`" must be specified.
- Ellipses ... denote that the preceding term may be repeated. For example, `[ a | b ] ...` means that you may have the sequence, "`b a a b a`".

2 Moving an Existing Database into an EDB Ark Cluster

You can use the pg_dump utility to create an archive that will recreate the objects in an existing Postgres database (schema, data, and associated database objects) in an EDB Ark cluster. The process consists of three basic steps:

1. Use the pg_dump utility on the source database to create a plain-text archive that contains the SQL commands needed to re-create the content of the existing database in an EDB Ark cluster.

2. Copy the archive to the target EDB Ark cluster.

3. Use the psql client on the target database to play back the SQL commands in the archive, recreating the content of the database.

You can use the pg_dumpall utility to move an entire cluster (data, schema information, and roles) to EDB Ark; for detailed information about using pg_dumpall, please see the Postgres documentation at:

http://www.postgresql.org/docs/9.5/static/app-pg-dumpall.html

The examples that follow demonstrate moving the Advanced Server sample database (edb) into an EDB Ark cluster.

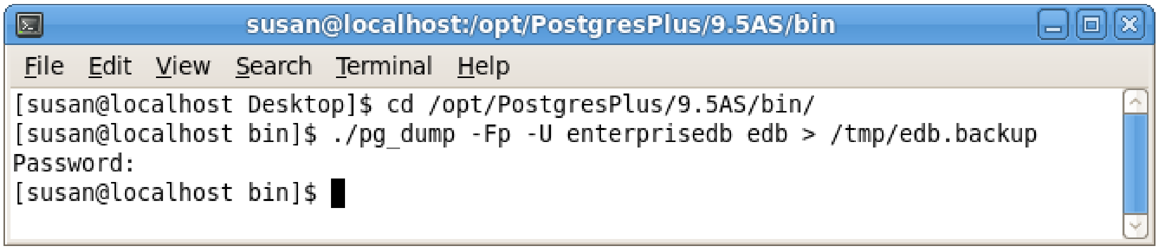
2.1 Using pg_dump to Create a Plain-Text Backup

The pg_dump utility can generate an archive that contains the SQL statements required to recreate a database or database object in a cluster that resides on OpenStack.

Then, use the pg_dump utility to create an archive that contains the commands required to recreate the database. By default, pg_dump is installed in the bin directory under your Postgres installation. If the bin directory is not in your path, specify the path to the bin directory when invoking pg_dump.

Include the -Fp flag to instruct pg_dump to format the output as a plain-text file, and the -U flag to specify the name of the database superuser (see Figure 2.1):

```
pg_dump -Fp -U db_superuser db_name > db.backup
```


susan@localhost:/opt/PostgresPlus/9.5AS/bin
File Edit View Search Terminal Help

```
[susan@localhost Desktop]$ cd /opt/PostgresPlus/9.5AS/bin/
[susan@localhost bin]$ ./pg_dump -Fp -U enterprisedb edb > /tmp/edb.backup
Password:
[susan@localhost bin]$ 
```

Figure 2.1 - Creating the pg_dump archive.

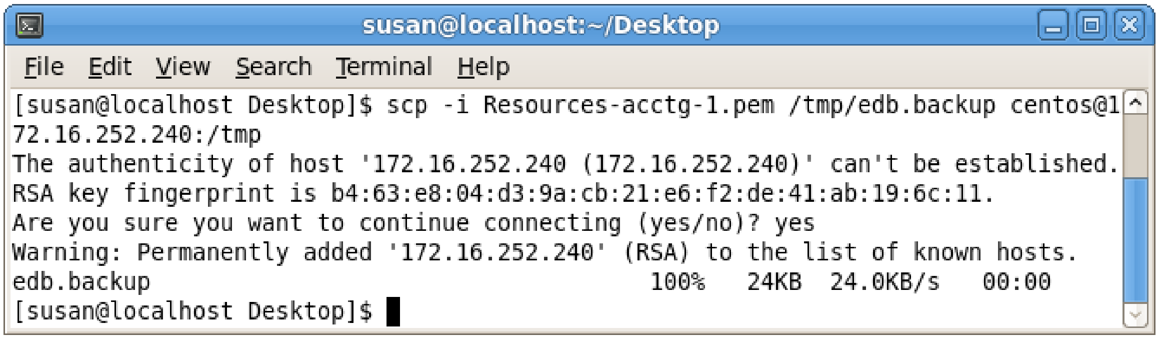
Where:

> *db_superuser* is the name of a Postgres database superuser.
>
> *db_name* is the name of the database that you wish to move to an EDB Ark cluster.
>
> *db.backup* is the complete path and name of the archive. Please note that you must have permission to write a file to the location specified.

If prompted, enter the password associated with the database superuser.

2.2 Moving the Archive to the EDB Ark Host

Please note: before using `ssh` or `scp` to connect to the Cloud cluster, an OpenStack administrator must modify the OpenStack security group for the cluster, opening port 22 for connections from your system.


susan@localhost:~/Desktop
File Edit View Search Terminal Help

```
[susan@localhost Desktop]$ scp -i Resources-acctg-1.pem /tmp/edb.backup centos@1
72.16.252.240:/tmp
The authenticity of host '172.16.252.240 (172.16.252.240)' can't be established.
RSA key fingerprint is b4:63:e8:04:d3:9a:cb:21:e6:f2:de:41:ab:19:6c:11.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.252.240' (RSA) to the list of known hosts.
edb.backup                                    100%   24KB  24.0KB/s   00:00
[susan@localhost Desktop]$
```

Figure 2.2 - Moving the archive to the Cloud.

Use the `scp` command to copy the archive to the master server in the EDB Ark cluster; include the `-i` option to specify the location of your `ssh` key (see Figure 2.2):

```
scp -i ssh_key_file file_name user_name@host_name:target
```

Where:

`ssh_key_file` specifies the complete path and name of the `ssh` key file. Use the `download SSH Key` icon on the `Clusters` tab of the EDB Ark console to download your ssh key file. After downloading the key file, you must modify the permissions on the file, restricting access to the key; use the command:

```
chmod 600 ssh_key_file
```

`file_name` specifies the archive name.

`user_name` specifies the name used to connect to the master node of the cluster.

`host_name` specifies the host name of the master node of the target cluster; the host name is located on the `Details` panel of the `Clusters` tab in the EDB Ark console.

`d` specifies the name of the target directory on the EDB Ark host.

2.3 Connecting to the EDB Ark Host with ssh

After moving the archive to the EDB Ark host, use `ssh` to connect to your EDB Ark cluster master node (see Figure 2.3):

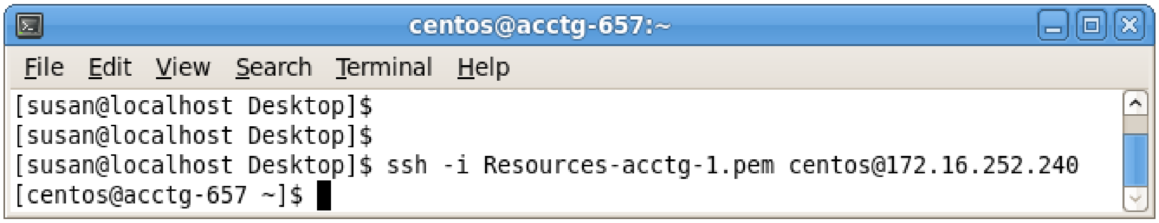
```
ssh -i ssh_key_file user@host_name
```

Where:

`ssh_key_file` specifies the complete path and name of the `ssh` key file.

`user` specifies the name of the connecting user.

`host_name` specifies the host name of the master node of the EDB Ark cluster; the host name is located on the `Details` panel of the `Clusters` tab in the EDB Ark console.



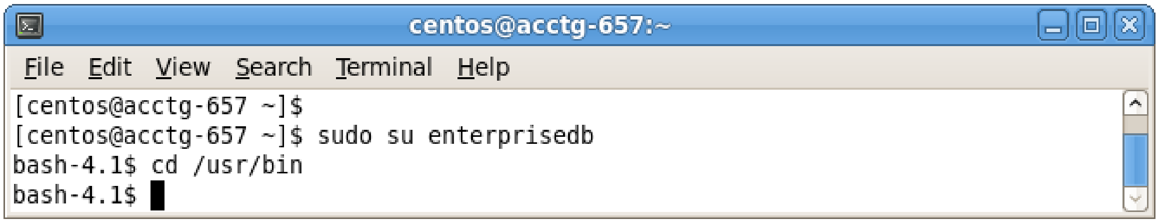
```
[susan@localhost Desktop]$
[susan@localhost Desktop]$
[susan@localhost Desktop]$ ssh -i Resources-acctg-1.pem centos@172.16.252.240
[centos@acctg-657 ~]$ 
```

Figure 2.3 - Connecting to EDB Ark cluster with ssh.

2.4 Using the psql Client to Recreate the Database

After connecting, assume the identity of the database superuser.

By default, the `createdb` utility and `psql` client are installed in the `bin` directory under your Postgres installation. If the `bin` directory is not in your path, you can navigate into the directory, or specify the path to the directory when invoking `psql` (see Figure 2.4).



```
[centos@acctg-657 ~]$
[centos@acctg-657 ~]$ sudo su enterprisedb
bash-4.1$ cd /usr/bin
bash-4.1$ 
```

Figure 2.4 – Navigate into the bin directory.

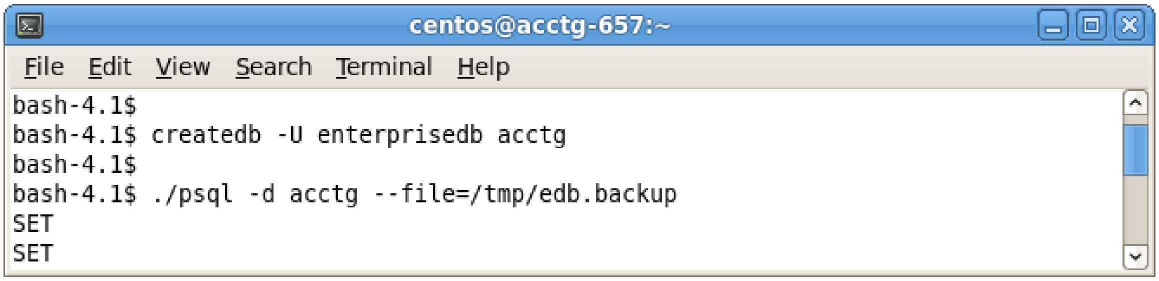
Before executing the commands that will recreate the database contents on the Cloud cluster, use the `createdb` client utility to create the target database:

```
createdb -U db_superuser database_name
```

Where:

`db_superuser specifies` the name of the database superuser. On an Advanced Server cluster, the default is `enterprisedb`; on a PostgreSQL cluster, the default is `postgres`.

`database_name` specifies the name of the database on the Cloud.



```
bash-4.1$
bash-4.1$ createdb -U enterprisedb acctg
bash-4.1$
bash-4.1$ ./psql -d acctg --file=/tmp/edb.backup
SET
SET
```

Figure 2.5 - Recreating the objects from the source database on the Cloud.

Then, invoke the `psql` client to recreate the database objects within the new database (see Figure 2.5).

```
psql -d database_name --file=db.backup
```

Where:

database_name specifies the name of the database that will contain the objects created by the archive.

db.backup is the name of the archive file.

Include:

the `-d` *target_db_name* flag to specify the name of the target database

the --`file=` psql keyword to specify the complete path and name of the archive file.

2.5 Confirming that the Move was Successful

You can use the `psql` client to connect to the EDB Ark and confirm that the contents of the database have been recreated in the new cluster (see Figure 2.6):

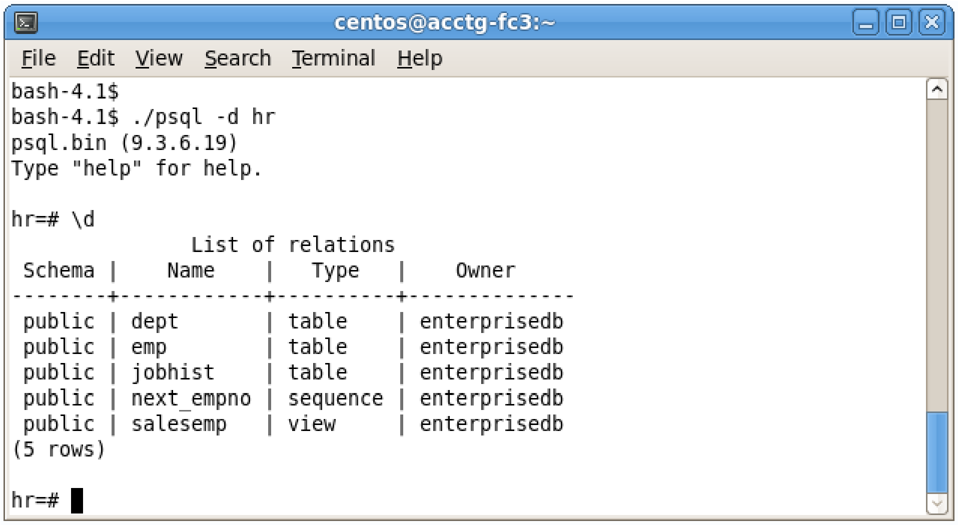
```
psql -U database_superuser -d db_name
```

Where:

db_superuser specifies the name of the database superuser. On an Advanced Server cluster, the default is `enterprisedb`; on a PostgreSQL cluster, the default is `postgres`.

db_name is the name of the target database.

Use the `\d` meta-command to view a list of database objects in the current database:



```
bash-4.1$
bash-4.1$ ./psql -d hr
psql.bin (9.3.6.19)
Type "help" for help.

hr=# \d
               List of relations
 Schema |    Name    |   Type   |    Owner
--------+------------+----------+-------------
 public | dept       | table    | enterprisedb
 public | emp        | table    | enterprisedb
 public | jobhist    | table    | enterprisedb
 public | next_empno | sequence | enterprisedb
 public | salesemp   | view     | enterprisedb
(5 rows)

hr=#
```

Figure 2.6 - Confirming that the move was successful.

To exit the `psql` client, enter `\q`; to exit the `ssh` session, type `exit` and `Return`.

For more information about using the `psql` client, or about using `pg_dump` or `pg_restore` to move an existing database into an EDB Ark cluster, please see the core documentation at:

For `pg_restore`: http://www.postgresql.org/docs/current/static/app-pgrestore.html

For `pg_dump`: http://www.postgresql.org/docs/current/static/app-pgdump.html