

1. Screenshot of AWS Console showing the volumes you have allocated on EBS.

The screenshot shows the AWS Management Console interface for the Elastic Block Store (EBS) Volumes page. The console displays a table of two EBS volumes. The first volume has ID vol-00549108cd30a9d05, size 1 GiB, type gp2, and is in the us-west-2a availability zone. The second volume has ID vol-0aa4cd33e9e6b49c1, size 8 GiB, type gp2, and is also in the us-west-2a availability zone. Both volumes are in the 'in-use' state. The console interface includes a sidebar with navigation options like EC2 Dashboard, INSTANCES, IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, and LOAD BALANCING. The top navigation bar shows the user is logged in as Khushali in the Oregon region.

Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability Zone	State	Alarm
	vol-00549108cd30a9d05	1 GiB	gp2	100 / 3000		September 25, 201...	us-west-2a	in-use	None
	vol-0aa4cd33e9e6b49c1	8 GiB	gp2	100 / 3000	snap-d465048a	September 25, 201...	us-west-2a	in-use	None

2. Output of Linux command “df” in instance after Step (2).

3. Output of “ls -l /data” after Step (3).

4. Output of “more -10 /data/pg_hba.conf” after Step (3).

```
bash-4.2$ more -10 /data/pg_hba.conf
# PostgreSQL Client Authentication Configuration File
# =====
#
# Refer to the "Client Authentication" section in the
# PostgreSQL documentation for a complete description
```

```
.ec2 — root@ip-172-31-34-249:~ — ssh -i ec2-glassfish-keypair.pem ec2-use...
[bash-4.2$ ls -l /data
total 92
drwx----- 6 postgres postgres 4096 Sep 26 05:17 base
drwx----- 2 postgres postgres 4096 Oct  3 18:21 global
drwx----- 2 postgres postgres 4096 Sep 26 02:32 pg_clog
-rw-r--r-- 1 postgres postgres 3700 Sep 26 04:55 pg_hba.conf
-rw----- 1 postgres postgres 1636 Sep 26 02:32 pg_ident.conf
drwx----- 2 postgres postgres 4096 Sep 26 05:00 pg_log
drwx----- 4 postgres postgres 4096 Sep 26 02:31 pg_multixact
drwx----- 2 postgres postgres 4096 Oct  3 18:20 pg_notify
drwx----- 2 postgres postgres 4096 Sep 26 02:31 pg_serial
drwx----- 2 postgres postgres 4096 Sep 26 02:31 pg_snapshots
drwx----- 2 postgres postgres 4096 Oct  3 18:42 pg_stat_tmp
drwx----- 2 postgres postgres 4096 Sep 26 02:32 pg_subtrans
drwx----- 2 postgres postgres 4096 Sep 26 02:31 pg_tblspc
drwx----- 2 postgres postgres 4096 Sep 26 02:31 pg_twophase
-rw----- 1 postgres postgres    4 Sep 26 02:31 PG_VERSION
drwx----- 3 postgres postgres 4096 Sep 26 02:32 pg_xlog
-rw-r--r-- 1 postgres postgres 16947 Sep 26 02:59 postgresql.conf
-rw----- 1 postgres postgres   45 Oct  3 18:20 postmaster.opts
-rw----- 1 postgres postgres   69 Oct  3 18:20 postmaster.pid
bash-4.2$
```

of this file. A short synopsis follows.

#

This file controls: which hosts are allowed to connect, how clients
are authenticated, which PostgreSQL user names they can use, which
databases they can access. Records take one of these forms:

#

local DATABASE USER METHOD [OPTIONS]

host DATABASE USER CIDR-ADDRESS METHOD [OPTIONS]

hostssl DATABASE USER CIDR-ADDRESS METHOD [OPTIONS]

hostnossl DATABASE USER CIDR-ADDRESS METHOD [OPTIONS]

#

(The uppercase items must be replaced by actual values.)

#

The first field is the connection type: "local" is a Unix-domain
socket,

"host" is either a plain or SSL-encrypted TCP/IP socket, "hostssl"
is an

SSL-encrypted TCP/IP socket, and "hostnossl" is a plain TCP/IP
socket.

#

DATABASE can be "all", "sameuser", "samerole", a database name, or
a comma-separated list thereof.

#

USER can be "all", a user name, a group name prefixed with "+", or

```

# a comma-separated list thereof. In both the DATABASE and USER
# fields
# you can also write a file name prefixed with "@" to include names
# from
# a separate file.
#
# CIDR-ADDRESS specifies the set of hosts the record matches.
# It is made up of an IP address and a CIDR mask that is an integer
# (between 0 and 32 (IPv4) or 128 (IPv6) inclusive) that specifies
# the number of significant bits in the mask. Alternatively, you can
# write
# an IP address and netmask in separate columns to specify the set of
# hosts.
#
# METHOD can be "trust", "reject", "md5", "password", "gss", "sspi",
# "krb5",
# "ident", "pam", "ldap" or "cert". Note that "password" sends
# passwords
# in clear text; "md5" is preferred since it sends encrypted
# passwords.
#
# OPTIONS are a set of options for the authentication in the format
# NAME=VALUE. The available options depend on the different
# authentication
# methods - refer to the "Client Authentication" section in the
# documentation
# for a list of which options are available for which authentication
# methods.
#
# Database and user names containing spaces, commas, quotes and other
# special
# characters must be quoted. Quoting one of the keywords "all",
# "sameuser" or
# "samerole" makes the name lose its special character, and just match
# a
# database or username with that name.
#
# This file is read on server startup and when the postmaster receives
# a SIGHUP signal. If you edit the file on a running system, you have
# to SIGHUP the postmaster for the changes to take effect. You can
# use
# "pg_ctl reload" to do that.

# Put your actual configuration here
# -----
#
# If you want to allow non-local connections, you need to add more
# "host" records. In that case you will also need to make PostgreSQL
# listen

```

```
# on a non-local interface via the listen_addresses configuration
parameter,
# or via the -i or -h command line switches.
#
```

```
# CAUTION: Configuring the system for local "trust" authentication
allows
# any local user to connect as any PostgreSQL user, including the
database
# superuser. If you do not trust all your local users, use another
# authentication method.
```

```
# TYPE  DATABASE  USER          CIDR-ADDRESS  METHOD

# "local" is for Unix domain socket connections only
local   all             postgres      trust
# IPv4 connections:
host    all             lord          0.0.0.0/0     md5
host    all             serf          0.0.0.0/0     md5
# IPv6 local connections:
host    all             all           ::1/128       md5
```

5. Output of “psql -U postgres -c ‘\du’” after Step (3)

```
[~bash-4.2$ psql -U postgres -c '\du'
                                List of roles
Role name | Attributes | Member of
-----+-----+-----
lord      | Superuser, Create role, Create DB | {}
postgres  | Superuser, Create role, Create DB, Replication | {}
serf      |          | {}
```

6. Output of “java -version” after Step (4).

```
ec2 — ec2-user@ip-172-31-34-249:~ — ssh -i ec2-glassfish-keypair.pem ec2...
[[ec2-user@ip-172-31-34-249 ~]$ java -version
java version "1.8.0_102"
Java(TM) SE Runtime Environment (build 1.8.0_102-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.102-b14, mixed mode)
[ec2-user@ip-172-31-34-249 ~]$
```

7. Output of “ls -l /usr/share/glassfish4” after Step (4).

```
ec2 — glassfish@ip-172-31-34-249:/root — ssh -i ec2-glassfish-keypair.pem...
[[ec2-user@ip-172-31-34-249 ~]$ sudo su -
Last login: Mon Oct  3 18:54:23 UTC 2016 on pts/0
Last failed login: Mon Oct  3 19:05:30 UTC 2016 on pts/0
There was 1 failed login attempt since the last successful login.
[[root@ip-172-31-34-249 ~]# su glassfish -
[[glassfish@ip-172-31-34-249 root]$ ls -l /usr/share/glassfish4
total 24
drwxr-xr-x  2 glassfish glassfish 4096 Aug 21  2014 bin
drwxr-xr-x 11 glassfish glassfish 4096 Aug 21  2014 glassfish
drwxr-xr-x  4 glassfish glassfish 4096 Aug 21  2014 javadb
drwxr-xr-x  5 glassfish glassfish 4096 Aug 21  2014 mq
drwxr-xr-x  4 glassfish glassfish 4096 Aug 21  2014 pkg
-rw-r--r--  1 glassfish glassfish 2788 Aug 21  2014 README.txt
[glassfish@ip-172-31-34-249 root]$
```

8. Screenshot of Glassfish Admin Console, showing Resources\JDBC\Connection Pools.

Home About... Logout Help

User: admin Role: domain1 Server: ec2-54-70-149-214.us-west-2.compute.amazonaws.com

GlassFish™ Server Open Source Edition

Total # of available updates : 45

Common Tasks

- Domain
- server (Admin Server)
- Clusters
- Standalone Instances
- Nodes
- Applications
- Lifecycle Modules
- Monitoring Data
- Resources
 - Concurrent Resources
 - Connectors
 - JDBC
 - JDBC Resources
 - JDBC Connection Pools**
 - JMS Resources
 - JNDI
 - JavaMail Sessions
 - Resource Adapter Configs
- Configurations
 - default-config
 - server-config
- Update Tool

JDBC Connection Pools

To store, organize, and retrieve data, most applications use relational databases. Java EE applications access relational databases through the JDBC API. Before an application can access a database, it must get a connection.

Pools (3)

New... Delete

Select	Pool Name	Resource Type	Classname	Description
<input type="checkbox"/>	DerbyPool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource	
<input type="checkbox"/>	TimerPool	javax.sql.XADataSource	org.apache.derby.jdbc.EmbeddedXADataSource	
<input type="checkbox"/>	serfdbPool	javax.sql.ConnectionPoolDataSource	org.postgresql.ds.PGConnectionPoolDataSource	

9. The administrator passwords chosen for Glassfish. No points for poor (insecure) passwords.

Master Password: M@ster13!

Administrator Password: N0http:

10. Video of local browser window showing the “Messages” application running in your instance (same DNS as above). Grader must be able to replicate the successful deployment of the app.

In the same folder where README was present there is Assignment-1 Video file which has video of application deployed and running.

password.csv - contains the password and URL Link for accessing.

credentials.csv - contains Secret Access key