

Mysql migration of BigDB from 5.5 to 5.7 on CentOS 7

Thursday, 26 July 2018 4:23 PM

TO MIGRATE A HUGE DB THE CREATED INSTANCE HAS SHORTAGE OF STORAGE. SO LET'S CREATE A NEW VOLUME FOR THE INSTANCE USING THE FOLLOWING STEPS

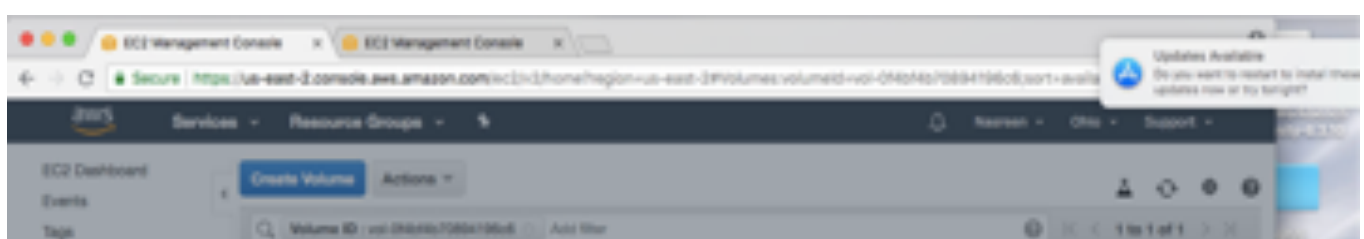
Click on "**Create volume**" menu and make changes in the requirements as needed. Here I've created a volume with the size 40G, then click on "**createvolume**" button.

The screenshot shows the AWS Management Console 'Create Volume' page. The browser address bar indicates the URL is <https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#CreateVolume:>. The page title is 'Create Volume'. The form includes the following fields:

- Volume Type:** General Purpose SSD (GP2)
- Size (GiB):** 40 (Min: 1 GiB, Max: 16384 GiB)
- IOPS:** 120 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)
- Availability Zone:** us-east-2b
- Throughput (MB/s):** Not applicable
- Snapshot ID:** Select a snapshot
- Encryption:** ☐ Encrypt this volume

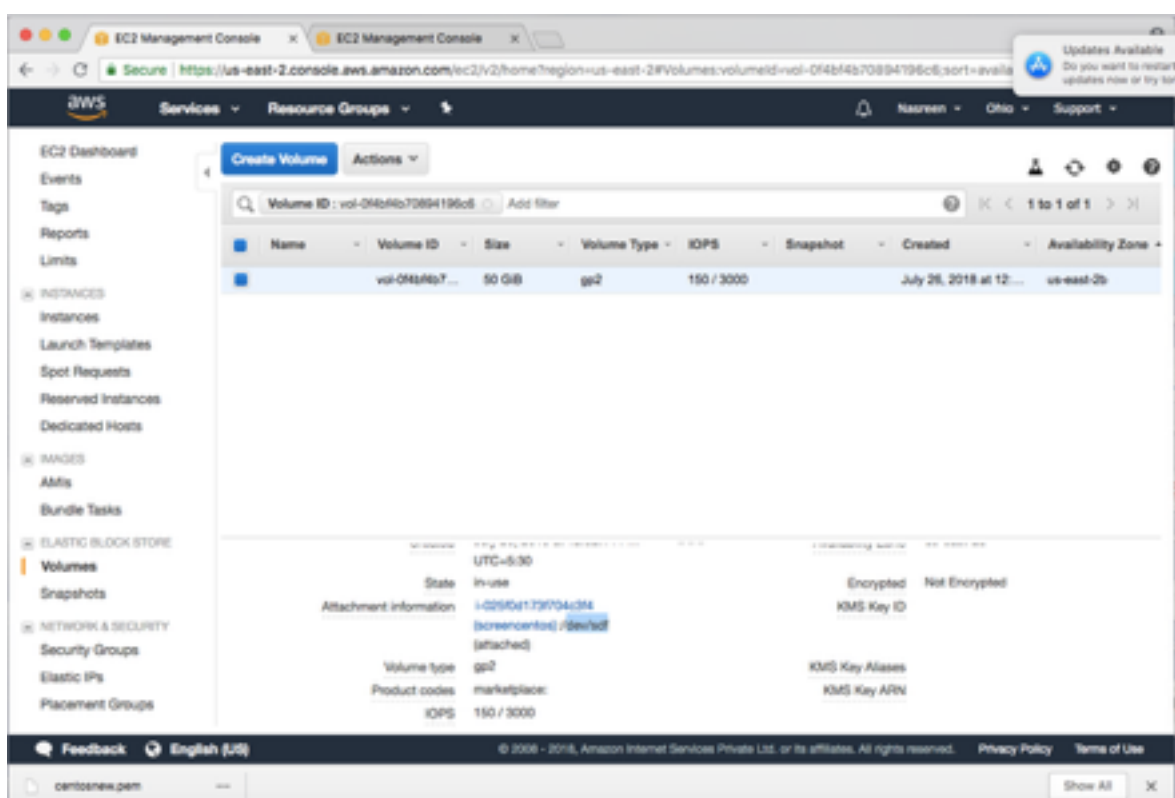
At the bottom, there is a section for 'Key' (127 characters maximum) and 'Value' (255 characters maximum). The footer of the console shows 'Feedback', 'English (US)', and copyright information for 2008-2018.

Attach the volume by right clicking on the running volume. And select the instance availability zone, then the instance that needs to be attached.





Go to the attached volume and copy the location



```
$ lsblk
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
└─xvda1     202:1    0   8G  0 part /
xvdf        202:80    0  50G  0 disk
```

AVAIL 202100 0 500 0 013K

```
$ sudo mkfs -t ext3 /dev/xvdf
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3276800 inodes, 13107200 blocks
655360 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
400 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
$ sudo su
```

Create a new directory

```
# mkdir /new
```

Now mount the directory to the drive

```
# sudo mount /dev/xvdf /new
```

```
# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvda1	8.0G	3.4G	4.6G	43%	/
devtmpfs	474M	0	474M	0%	/dev
tmpfs	496M	0	496M	0%	/dev/shm
tmpfs	496M	13M	483M	3%	/run
tmpfs	496M	0	496M	0%	/sys/fs/cgroup
tmpfs	100M	0	100M	0%	/run/user/1000
/dev/xvdf	50G	52M	47G	1%	/new

The directory is mounted successfully

the directory is mounted successfully

```
# cp -R /var/lib/mysql /new
# cd /var/lib
```

```
#cp -R /var/lib/mysql /new
```

```
# cd /var/lib
```

```
# ls
```

```
alternatives  dhclient  machines  nfs  rpm  systemd
authconfig    games     misc      os-prober  rpm-state tuned
chrony         gssproxy  mysql     polkit-1   rsyslog  yum
cloud          initramfs mysql-files postfix    selinux
dbus           logrotate mysql-keyring rpcbind    stateless
```

```
# mv mysql mysql-back
```

```
# cd /new
```

```
# ls
```

```
lost+found  mysql
```

```
# chown -R mysql:mysql /new/mysql/
```

```
# chown -R mysql:mysql /new/mysql
```

```
# ln -s /new/mysql/ /var/lib/
```

```
# chown -R mysql:mysql /var/lib/mysql
```

```
# cd ..
```

```
# cd /var/lib
```

```
# ls
```

```
alternatives  dhclient  machines  mysql-keyring  rpcbind  stateless
authconfig    games     misc      nfs            rpm      systemd
chrony         gssproxy  mysql     os-prober     rpm-state tuned
cloud          initramfs mysql-back  polkit-1      rsyslog  yum
dbus           logrotate mysql-files postfix        selinux
```

```
[root@ip-172-31-31-16 lib]# ll
```

```
total 8
```

```
drwxr-xr-x. 2 root root    55 Jun  5 14:09 alternatives
drwx-----. 3 root root    18 Jun  5 14:12 authconfig
drwxr-xr-x. 2 chrony chrony  19 Jul 26 06:33 chrony
drwxr-xr-x. 8 root root   105 Jul 26 05:27 cloud
drwxr-xr-x. 2 root root     6 Apr 11 07:23 dbus
drwxr-xr-x. 2 root root    34 Jul 26 05:27 dhclient
drwxr-xr-x. 2 root root     6 Apr 11 04:59 games
drwxr-xr-x. 4 root root    55 Jul 26 05:27 gssproxy
drwxr-xr-x. 2 root root     6 Apr 11 07:30 initramfs
drwxr-xr-x. 2 root root    30 Jul 26 06:31 logrotate
drwx-----. 2 root root     6 Jun  5 14:08 machines
drwxr-xr-x. 2 root root    37 Jul 26 05:27 misc
lrwxrwxrwx. 1 mysql mysql    11 Jul 26 07:30 mysql -> /new/mysql/
```

```

drwxr-x--x. 6 mysql mysql 4096 Jul 26 05:55 mysql-back
drwxr-x---. 2 mysql mysql 6 Jul 26 05:44 mysql-files
drwxr-x---. 2 mysql mysql 6 Mar 4 14:06 mysql-keyring
drwxr-xr-x. 5 root root 105 Jun 5 14:09 nfs
drwxr-xr-x. 2 root root 6 Nov 6 2016 os-prober
drwxr-x---. 3 root polkitd 28 Jun 5 14:08 polkit-1
drwx-----. 2 postfix root 25 Jul 26 05:27 postfix
drwx-----. 2 rpc rpc 6 Apr 11 05:13 rpcbind
drwxr-xr-x. 2 root root 4096 Jul 26 05:35 rpm
drwxr-xr-x. 2 root root 6 Apr 11 04:59 rpm-state
drwx-----. 2 root root 29 Jul 26 07:17 rsyslog
drwxr-xr-x. 2 root root 6 Apr 11 16:10 selinux
drwxr-xr-x. 4 root root 35 Jun 5 14:08 stateless
drwxr-xr-x. 4 root root 56 Jun 5 14:08 systemd
drwxr-xr-x. 2 root root 6 Apr 11 02:57 tuned
drwxr-xr-x. 6 root root 80 Jul 26 05:43 yum

```

```
# service mysqld stop
```

```
Redirecting to /bin/systemctl stop mysqld.service
```

In centOS 7 the selinux enforce has to disabled to avoid server start issues

```
# vi /etc/sysconfig/selinux
```

```

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are pro
protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

~
~
~
~
~
~
~
"/etc/sysconfig/selinux" 14L, 545C

```

The give command as follows:

```
# setenforce 0
```

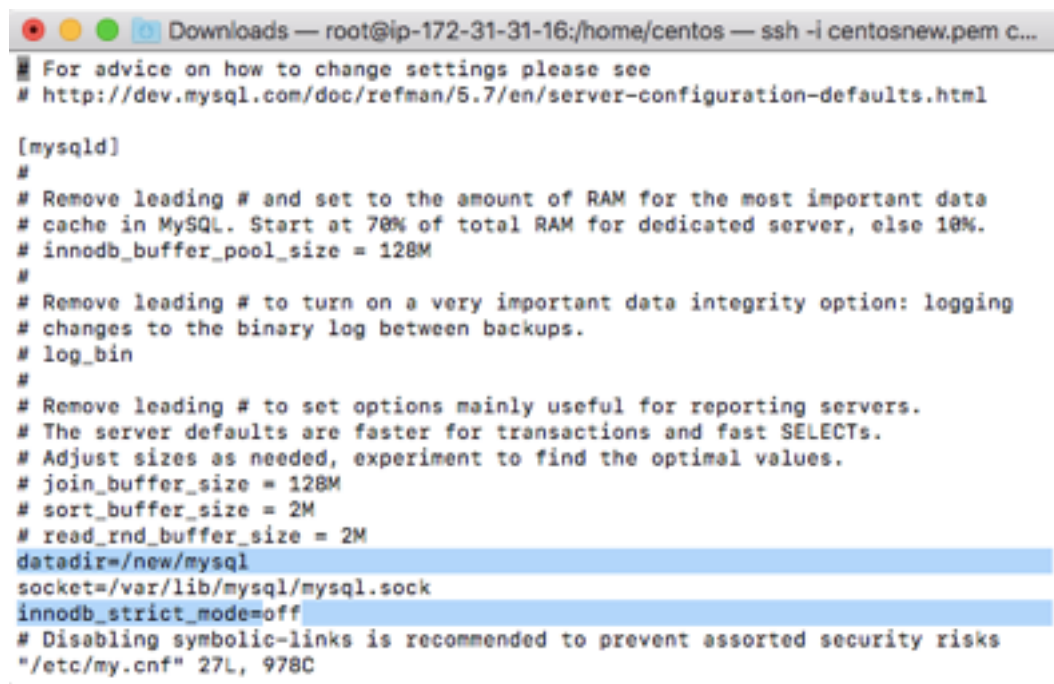
```
# sestatus
```

```
SELinux status: enabled
```

```
SELinuxfs mount:           /sys/fs/selinux
SELinux root directory:    /etc/selinux
Loaded policy name:        targeted
Current mode:              permissive
Mode from config file:     disabled
Policy MLS status:         enabled
Policy deny_unknown status: allowed
Max kernel policy version: 31
```

Then make changes in mysql config to switchoff the strict mode and change the linked datadir in it.

```
# vi /etc/my.cnf
```



```
Downloads — root@ip-172-31-31-16:/home/centos — ssh -i centosnew.pem c...
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/5.7/en/server-configuration-defaults.html

[mysqld]
#
# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
# innodb_buffer_pool_size = 128M
#
# Remove leading # to turn on a very important data integrity option: logging
# changes to the binary log between backups.
# log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/new/mysql
socket=/var/lib/mysql/mysql.sock
innodb_strict_mode=off
# Disabling symbolic-links is recommended to prevent assorted security risks
"/etc/my.cnf" 27L, 978C
```

```
# systemctl restart mysqld.service
```

```
# systemctl status mysqld.service
```

● mysqld.service – MySQL Server

Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)

Active: **active (running)** since Thu 2018-07-26 07:37:19 UTC; 22s ago

Docs: man:mysqld(8)

<http://dev.mysql.com/doc/refman/en/using-systemd.html>

Process: 12017 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid \$MYSQLD_OPTS (code=exited, status=0/SUCCESS)

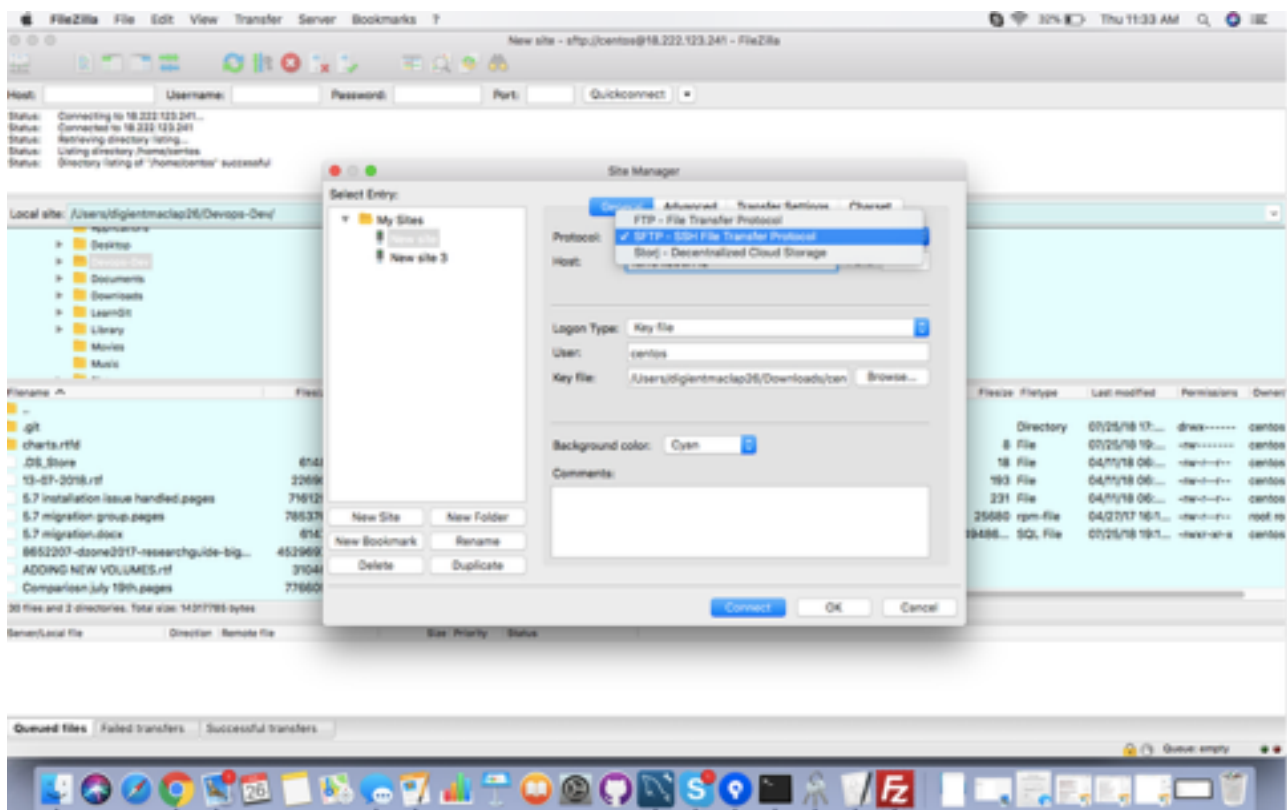
```
Process: 11996 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited,
status=0/SUCCESS)
Main PID: 12020 (mysqld)
  CGroup: /system.slice/mysqld.service
          └─12020 /usr/sbin/mysqld --daemonize --pid-
file=/var/run/mysqld/my...
```

Jul 26 07:37:17 ip-172-31-31-16.us-east-2.compute.internal systemd[1]:
Starti...

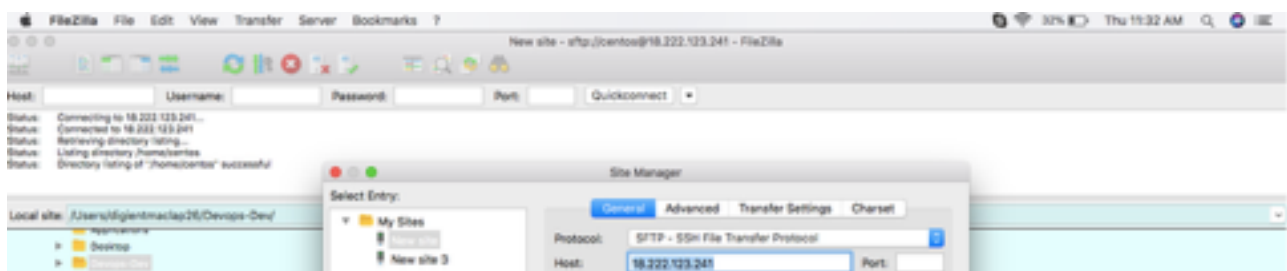
Jul 26 07:37:19 ip-172-31-31-16.us-east-2.compute.internal systemd[1]:
Starte...

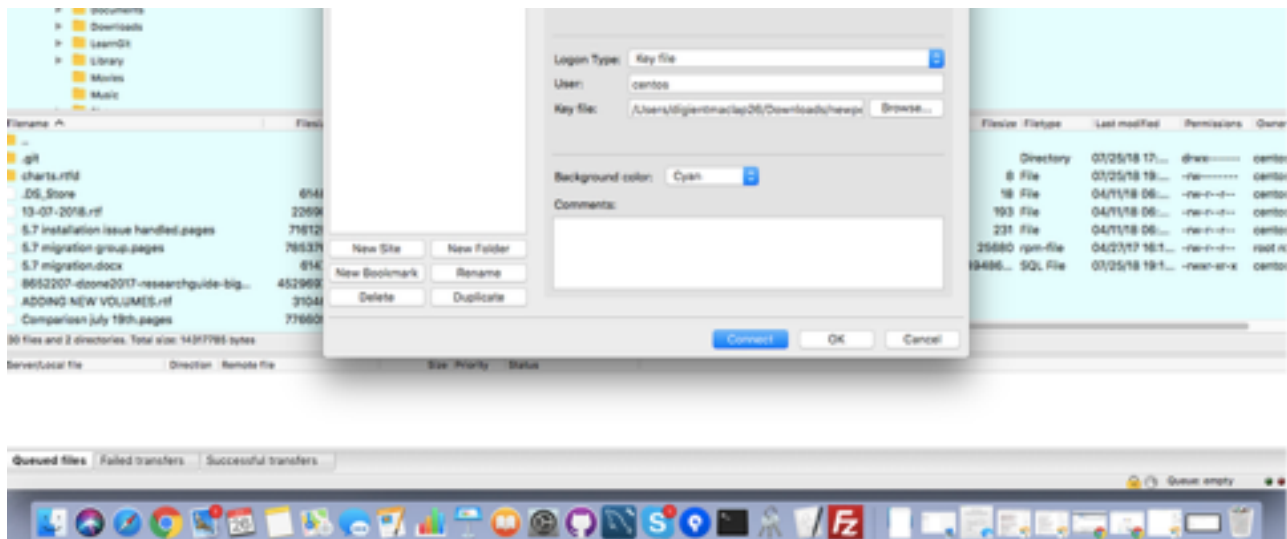
Hint: Some lines were ellipsized, use -l to show in full.

Now upload the db into the server in file zilla to import

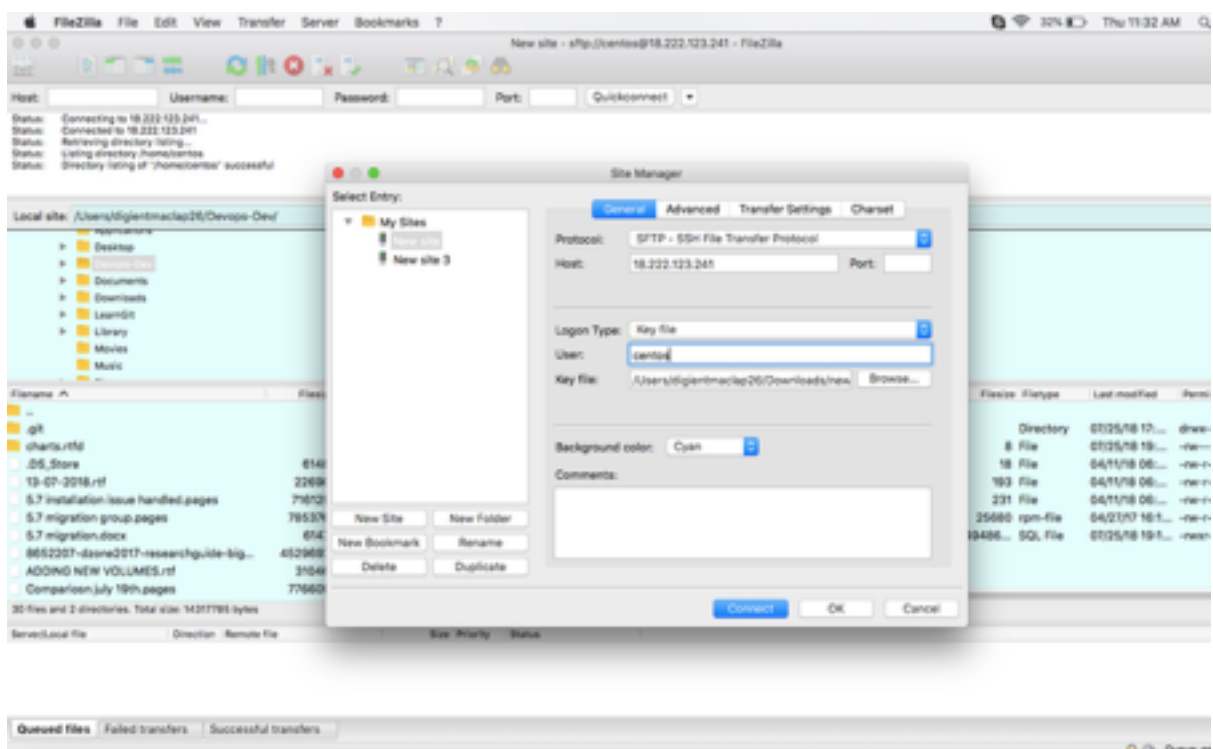


Connect the host



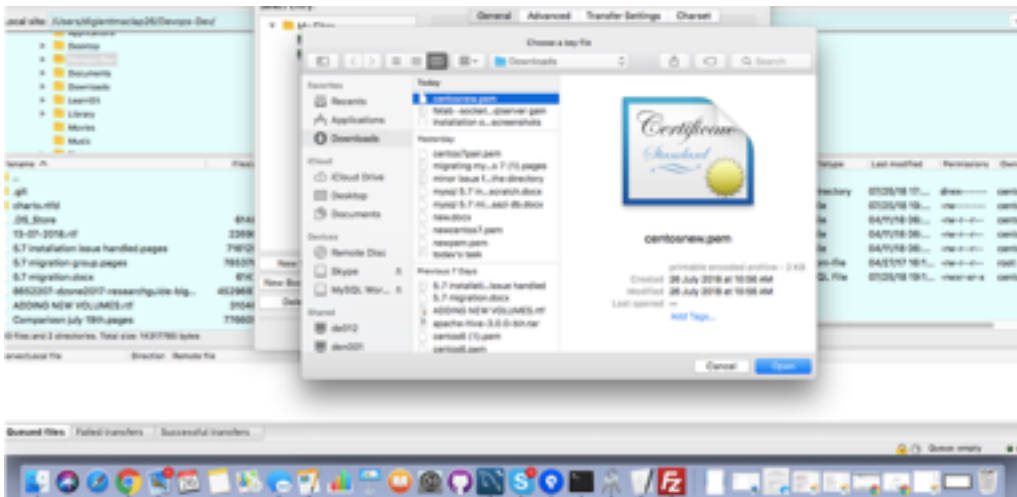


Make the following changes

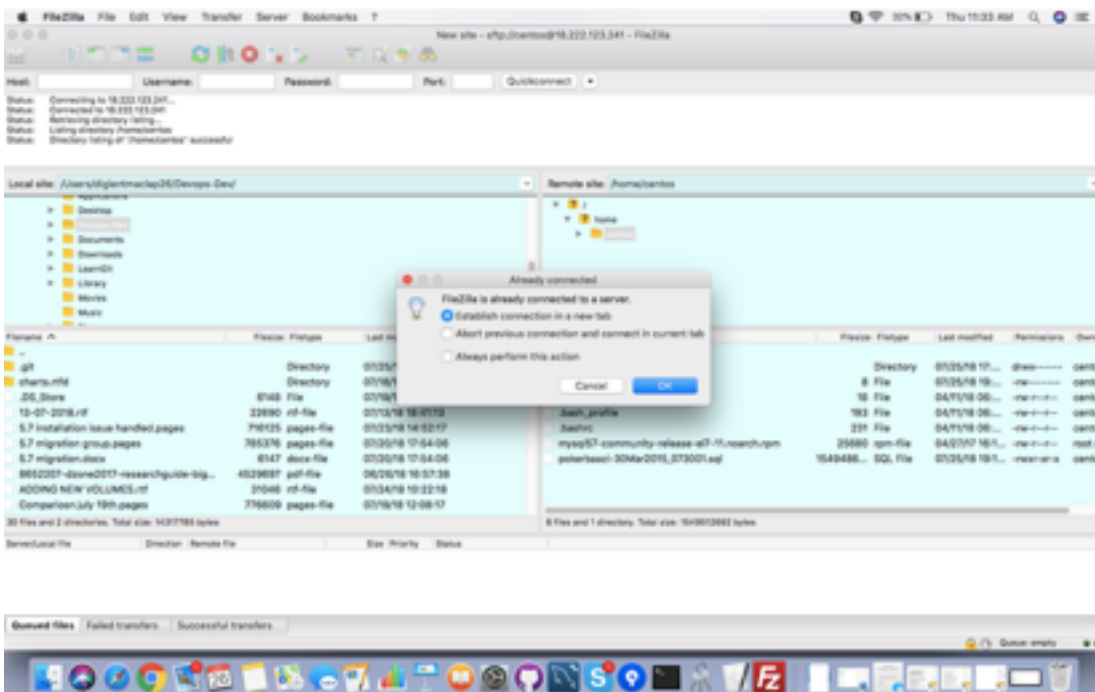


Attach the accesskey- pem file.

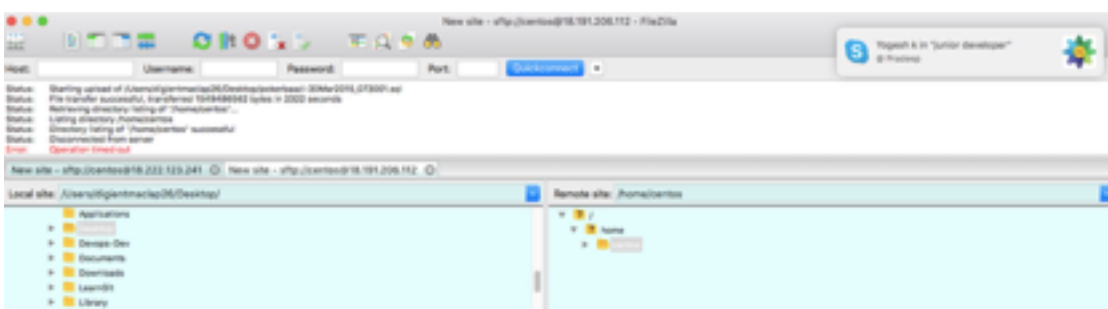


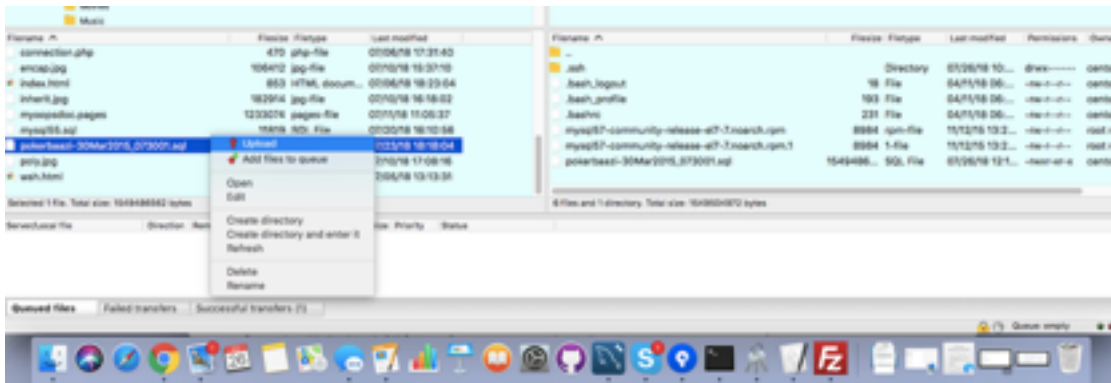


Click on Ok button



Upload the db from desktop- where you have stored it





After uploading,

Migrate the db to the drive using the following command

```
# mysql -uroot -p newmig < /home/centos/pokerbaazi-30Mar2015_073001.sql
```

```
# exit
```

```
$ mysql -uroot -p
```

Finally view the migrated db and its table in mysql