Change Description:

Upgrade MySQL to 5.6.33 and kernel packages upgrade to 2.6.32-642.4.2 for servers developer1.db.scl3.mozilla.com and developer2.db.scl3.mozilla.com.

- Kernel updates require a reboot
- MySQL update is for RCE CVE-2016-6662 - mandated security patch by Mozilla Enterprise Security

Implementation Window Date/Time/Timezone:

Downtime Required (Y/N)? Y
Expected length? 75 minutes
Stage: 10/06/2016 - 04:00 pm EDT
Production: 10/08/2016 - 11:30 am EDT

Detailed Communication Plan:

1) Inform in #mdn and #moc the start of maintenance channel steps under work.
2) Confirm with Matt before start working on it.

Approved By:

<need review>

Reviewed By:

<need review>

Action/Change Plan:

```
Work order is:
1 - developer2.db.scl3.mozilla.com kernel/mysql update
2 - Fail over
3 - developer1.db.scl3.mozilla.com kernel/mysql update
```

1) **Make sure developer2 is NOT in the load balancer.**
Reference for Mozilla instructions: https://mana.mozilla.org/wiki/display/BIDW/How+to+manually+fail+over+a+MySQL+master

   a) Login to the load balancer at https://www.zlb.ops.scl3.mozilla.com:9090/apps/zxtm/login.cgi with LDAP credentials
   b) Go to stage-ro-db pool at https://10.22.8.209:9090/apps/zxtm/?name=developer-ro-db&section=Virtual%20Servers%3AEdit
      i) Set 10.22.70.25:3306 to “active” and 10.22.70.26 to “disabled”

UPDATE THE SHEET:
https://docs.google.com/spreadsheets/d/1GsS-2jkVmMeaA2xcONW-E9ch41BL_O2PL_iLnPBjAUc/edit#gid=0
Set “developer2” to “WORKING”
Inform in #db-alerts that you are starting to work in the server.

2) **Make sure you are on the target host and in the expected window in the sheet.**

```
hostname   -- should return “developer2.db.scl3.mozilla.com”
ifconfig
rpm -qa |grep kernel
rpm -qa |grep libtiff
```

Take note of package versions, as in roll back we may want to downgrade them.

3) **Set downtime in nagios (step inside Mana)**
Example for IRC chat #db-alerts, here We downtime the server to be worked and the slave to avoid alarm of the replication.

```
nagios-scl3: downtime developer2.db.scl3.mozilla.com 75m upgrade kernel and mysql packages for Bugs 1302122  1292084
nagios-scl3: downtime developer1.db.scl3.mozilla.com:MySQL Replication 75m upgrade kernel and mysql packages for Bugs 1302122  1292084
```

Reference in : https://mana.mozilla.org/wiki/display/BIDW/How+to+manually+fail+over+a+MySQL+master#HowtomanuallyfailoveraMySQLmaster-Nagiosdownt

4) **In developer2.db.scl3.mozilla.com Shutdown Instance**
Shutdown DB and Apply the packages upgrade
Shutdown databases as necessary.

```
service mysqld stop
```

5) **Open Mana page as reference and Update databases.pp file**
Reference: https://mana.mozilla.org/wiki/display/BIDW/MySQL+5.6.17+to+5.6.30+Upgrade

Update MySQL in puppet - in db_mysql/databases.pp:
From:

```
$mysql_package_type = 'mysql56_30'
```

To:

```
$mysql_package_type = $::fqdn ? { /^developer2/ => 'mysql56_33', default => 'mysql56_30', }
```

6) **Commit the change, and note this revision number (e.g. "Committed revision 117965.")**

```
svn commit -m "Upgrading host developer2 Bugs 1292084 and 1302122"
```

7) **Update packages for Kernel and others as the removal of packages from mysql may break yum-wrapper.**
In server developer2.db.scl3.mozilla.com:
In the case of packages can be applied with application running, this is fine to do anytime. For kernel, kernel-devel and libtiff is the case.

```
yum-wrapper update kernel kernel-devel libtiff kernel-headers perf
```

```
Make sure grub.conf was updated (if not, it is necessary to remove and install packages with yum (and specific version) instead of update):
cat /boot/grub/grub.conf
```

8) **Remove the existing MySQL packages:**
In server developer2.db.scl3.mozilla.com:

```
yum-wrapper remove mysql-community-libs-5.6.30-2.el6.x86_64 mysql-community-common-5.6.30-2.el6.x86_64 mysql-community-client-5.6.30-2.el6.x86_64 mysql-community-libs-compat-5.6.30-2.el6.x86_64
mysql-community-server-5.6.30-2.el6.x86_64


OK to remove the following packages, they will be reinstated later (during puppet runs):
 collectd-mysql
 cronie
 cronie-anacron
 crontabs
 nagios-plugins-all
 nagios-plugins-mailq
 nagios-plugins-mysql
 percona-toolkit
 percona-xtrabackup
 perl-DBD-MySQL
 postfix
 Sysstat
```

9) **Reboot the machine; verify**
In server developer2.db.scl3.mozilla.com:

```
reboot
```

After the machine comes back up, verify that the kernel update was successful; “uname -a” should show the new kernel. (should take 5 min or less, contact Matt if it isn't coming up after 5 mins)

10) **Run puppet to upgrade MySQL**
In server developer2.db.scl3.mozilla.com:
Run puppet:

```
/usr/local/sbin/puppetctl run
```

Make sure that the version is equal to or greater than the revision number you noted above (the last "Info" line should be something like):

```
Info: Applying configuration version '117965'
```

Check the status with:

```
/usr/local/sbin/puppetctl status
```

11) **Run the puppet again:**
In server developer2.db.scl3.mozilla.com:
Run puppet:

```
/usr/local/sbin/puppetctl run
```

12) **Edit the /etc/my.cnf to turn of binlogging. Comment out log-bin, expire_log_days, log_slave_updates. Check /var/lib/mysql**
In server developer2.db.scl3.mozilla.com:
The cnf files need to have bin logs entries off for the mysql_upgrade work.
Comment:

```
log-bin
log_slave_updates
expire_log_days
```

Add:

```
skip-slave-start
```

`/var/lib/mysql`needs to be a symbolic link pointing to `/data/mysql`

```
ln -s /data/mysql /var/lib/mysql
```

13) **Start mysql**
In server developer2.db.scl3.mozilla.com:

```
service mysqld start
```

on backup hosts:

```
/data/backups/bin/start-all-dbs
```

(check logs, /var/lib/mysql/mysql.err)

14) **Run mysql_upgrade**
In server developer2.db.scl3.mozilla.com:

```
mysql_upgrade
```

15) **While this is running, run puppet, and there should be no errors - it should put binlogging back into /etc/my.cnf:**
In server developer2.db.scl3.mozilla.com:

```
/usr/local/sbin/puppetctl run
```

16) **RESTART MYSQL AND Check replication and status**
In server developer2.db.scl3.mozilla.com:

```
service mysqld restart
```

Log into server and run:

```
show slave status\G
```

17) **Check nagios is all OK except db uptime**
(db uptime will go green after 30 minutes, no need to wait for that check to go green)

```
nagios-scl3: status developer2.db.scl3.mozilla.com:MySQL
```

-- FAIL OVER TO WORK IN THE developer1 ---

18) **Set downtime in nagios (step inside Mana)**
Example for IRC chat #db-alerts, here We downtime the server to be worked and the slave to avoid alarm of the replication.

```
nagios-scl3: downtime developer1.db.scl3.mozilla.com 75m upgrade kernel and mysql packages for Bugs 1302122  1292084
nagios-scl3: downtime developer2.db.scl3.mozilla.com:MySQL Replication 75m upgrade kernel and mysql packages for Bugs 1302122  1292084
```

Reference in : https://mana.mozilla.org/wiki/display/BIDW/How+to+manually+fail+over+a+MySQL+master#HowtomanuallyfailoveraMySQLmaster-Nagiosdowntime

UPDATE THE SHEET:
https://docs.google.com/spreadsheets/d/1GsS-2jkVmMeaA2xcONW-E9ch41BL_O2PL_iLnPBjAUc/edit#gid=0
Set “developer1” to “WORKING”
Inform in #db-alerts that you are starting to work in the server.

19) **Change Puppet**
Change puppet so the "want_cron" variable in the "checksums" block (and any other block) is "true" on the soon-to-be master, and "false" on the current master. Look for and change all instances of "want_cron" unless there are notes not to.Change puppet so the "want_cron" variable in the "checksums" block (and any other block) is "true" on the soon-to-be master, and "false" on the current master. Look for and change all instances of "want_cron" unless there are notes not to.
To:

```
want_cron =>  $::fqdn ? { /^developer2/ => true, default   => false },
```

Change puppet so the "server_role" variable is "master" on the soon-to-be master, and "slave" on the current master. This is in trunk/manifests/nodes, are in the databases.pp file
To:

```
server_role => $::fqdn ? { /^developer2/ => 'master', default=> 'slave', },
```

```
svn commit -m "Upgrading host developer1 Bugs 1292084 and 1302122"
```

Have a Mozilla data team person (mpressman) change the nagios checks:
RESUME HERE: Run "/usr/local/sbin/puppetctl run" on both the soon-to-be and current masters (make sure it runs the revision number of your commit)

- Run "/usr/bin/pt-config-diff localhost /etc/my.cnf" on both machines.
  - Puppet changed the /etc/my.cnf.
  - Set read_only to OFF in developer2, and set it to OFF in developer1. AFTER fail over we set to ON in developer1.

20) **Fail over The Load balancer**
Open the following URLS, for the VIPs:
https://10.22.8.209:9090/apps/zxtm/?name=developer-ro-db&section=Virtual%20Servers%3AEdit - Read Only VIP
https://10.22.8.209:9090/apps/zxtm/?name=developer-rw-db&section=Virtual%20Servers%3AEdit - Read Write VIP

Log in and go to the stage-rw-db:

- Move Read Only VIP to developer2.db.scl3.mozilla.com which is 10.22.70.26:3306
- Set the current master (aka 10.22.70.25:3306) to "Draining" -> Save and click "Update"
- On developer1, use "show full processlist;" to see the number of active/sleeping connections for about roughly 15 seconds.
- After about 15 seconds, check the number of remaining connections. If only slave threads, nagios daemon, collectd and newrelic remain, move to the next step. If several to many non-slave threads still exist, stop the mysql service on the master with a normal mysql shutdown.
- In Zeus, set the developer1 (10.22.70.25:3306) to "Disabled" and the developer2 (10.22.70.26:3306) to "Active"
  Set read_only on in mysql instance at developer1 (10.22.70.25:3306).
- Triple check the read_only variable (you should have double checked it above) - it should be ON for developer1 and OFF for developer2.

-- Kernel and MySQL upgrade on developer1 ---

21) **In developer1.db.scl3.mozilla.com Shutdown Instance**
Shutdown DB and Apply the packages upgrade
Shutdown databases as necessary.

```
service mysqld stop
```

22) **Open Mana page as reference and Update databases.pp file**
Reference: https://mana.mozilla.org/wiki/display/BIDW/MySQL+5.6.17+to+5.6.30+Upgrade

Update MySQL in puppet - in db_mysql/databases.pp change:

From:

```
$mysql_package_type = $::fqdn ? { /^developer2/ => 'mysql56_33', default => 'mysql56_30', }
```

To:

```
$mysql_package_type = 'mysql56_33'
```

23) **Commit the change, and note this revision number (e.g. "Committed revision 117965.")**

```
svn commit -m "Upgrading host developer1 Bugs 1292084 and 1302122"
```

24) **Update packages for Kernel and others as the removal of packages from mysql may break yum-wrapper.**
In server developer1.db.scl3.mozilla.com:
In the case of packages can be applied with application running, this is fine to do anytime. For kernel, kernel-devel and libtiff is the case.

```
yum-wrapper update kernel kernel-devel libtiff kernel-headers perf

Make sure grub.conf was updated (if not, it is necessary to remove and install packages with yum (and specific version) instead of update):
cat /boot/grub/grub.conf
```

25) **Back on the host, remove the existing MySQL packages:**
In server developer1.db.scl3.mozilla.com:

```
yum-wrapper remove mysql-community-libs-5.6.30-2.el6.x86_64 mysql-community-common-5.6.30-2.el6.x86_64 mysql-community-client-5.6.30-2.el6.x86_64 mysql-community-libs-compat-5.6.30-2.el6.x86_64
mysql-community-server-5.6.30-2.el6.x86_64


OK to remove the following packages, they will be reinstated later (during puppet runs):
 collectd-mysql
 cronie
 cronie-anacron
 crontabs
 nagios-plugins-all
 nagios-plugins-mailq
 nagios-plugins-mysql
 percona-toolkit
 percona-xtrabackup
 perl-DBD-MySQL
 postfix
 Sysstat
```

26) **Reboot the machine; verify**

```
reboot
```

When the machine comes back up - Verify that the kernel update was successful; “uname -a” should show the new kernel. (should take 5 min or less, contact Matt if it isn't coming up after 5 mins)

27) **Run puppet to upgrade MySQL**

`In server` developer1.db.scl3.mozilla.com:

Run puppet:

```
/usr/local/sbin/puppetctl run
```

Make sure that the version is equal to or greater than the revision number you noted above (the last "Info" line should be something like):

```
Info: Applying configuration version '117965'
```

Check the status with:

```
/usr/local/sbin/puppetctl status
```

28) **Run the puppet again:**

`In server` developer1.db.scl3.mozilla.com:

Run puppet:

```
/usr/local/sbin/puppetctl run
```

29) **Edit the /etc/my.cnf to turn of binlogging. Comment log-bin, expire_log_days, log_slave_updates. Check /var/lib/mysql**

`In server` developer1.db.scl3.mozilla.com:

The cnf files need to have bin logs entries off for the mysql_upgrade work.

Comment:

```
log-bin
log_slave_updates
expire_log_days
```

Add:

```
skip-slave-start
```

`/var/lib/mysql`needs to be a symbolic link pointing to `/data/mysql`

```
ln -s /data/mysql /var/lib/mysql
```

30) **Start mysql**

`In server` developer1.db.scl3.mozilla.com:

```
service mysqld start
```

(check logs, /var/lib/mysql/mysql.err)

31) **Run mysql_upgrade**

`In server` developer1.db.scl3.mozilla.com:

```
mysql_upgrade
```

32) **While this is running, run puppet, and there should be no errors - it should put binlogging back into /etc/my.cnf:**
`In server` developer1.db.scl3.mozilla.com:

```
/usr/local/sbin/puppetctl run
```

33) **RESTART MYSQL AND Check replication and status**
`In server` developer1.db.scl3.mozilla.com:

```
service mysqld restart
```

Log into server and run:

```
show slave status\G
```

34) **Check nagios is all OK except db uptime**
(db uptime will go green after 30 minutes, no need to wait for that check to go green)

```
nagios-scl3: status developer1.db.scl3.mozilla.com:MySQL
```

35) **In the spreadsheet, update 'date work' and 'dba' columns**
Take note of the approximate time work was done and update the bug with it. Include hour:minute if possible. Update status to FINISHED

36) **Move back RO VIP**

- **a)** Login to the load balancer at https://www.zlb.ops.scl3.mozilla.com:9090/apps/zxtm/login.cgi with LDAP credentials
- **b)** Go to stage-ro-db pool at https://10.22.8.209:9090/apps/zxtm/?name=developer-ro-db&section=Virtual%20Servers%3AEdit
- c) Set 10.22.70.26:3306 to “active” and 10.22.70.25 to “disabled”

developer2 is the master, developer1 is the slave, both are upgraded, and slaving appropriately.

Engineers can do their schema updates on the stage site; the plan is to put the stage site into “hardhat” mode so there should be no need to stop any slaves or change the load balancer.

RollBack Plan:

Rollback need to be coordinated with Matt.
The rollback can be of only one package, kernel, mysql or all together. This need to be considered at timeof roll back.

For Kernel downgrade: Update grub.conf to use previous kernel
For MySQL: Use mysql_upgrade with downgraded package

```
Work order is:
1 - developer1.db.scl3.mozilla.com kernel/mysql update
2 - Fail over
3 - developer2.db.scl3.mozilla.com kernel/mysql update
```

37) Make sure you are on the target host and in the expected window in the sheet.

```
hostname -- should return “developer1.db.scl3.mozilla.com”
ifconfig
rpm -qa |grep kernel
rpm -qa |grep libtiff
```

Take note of packages, as in roll back we may want to downgrade them.

38) **Failover is needed, RO vip first.**
Reference for Mozilla instructions: https://mana.mozilla.org/wiki/display/BIDW/How+to+manually+fail+over+a+MySQL+master

a) Login to the load balancer at https://www.zlb.ops.scl3.mozilla.com:9090/apps/zxtm/login.cgi with LDAP credentials
b) Go to stage-ro-db pool at https://10.22.8.209:9090/apps/zxtm/?name=developer-ro-db&section=Virtual%20Servers%3AEdit

Set 10.22.70.26:3306 to “active” and 10.22.70.25 to “disabled”

39) **Set downtime in nagios (step inside Mana)**
Example for IRC chat #db-alerts, here We downtime the server to be worked and the slave to avoid alarm of the replication.

```
nagios-scl3: downtime developer1.db.scl3.mozilla.com 75m downgrade kernel and mysql packages for Bugs1302122  1292084
nagios-scl3: downtime developer2.db.scl3.mozilla.com:MySQL Replication 75m downgrade kernel and mysql packages for Bugs1302122  1292084
```

Reference in : https://mana.mozilla.org/wiki/display/BIDW/How+to+manually+fail+over+a+MySQL+master#HowtomanuallyfailoveraMySQLmaster-Nagiosdownt

40) **In developer1.db.scl3.mozilla.com Shutdown Instance**
Shutdown DB and Apply the packages upgrade
Shutdown databases as necessary.

```
service mysqld stop
```

41) **Open Mana page as reference and Update databases.pp file**
Reference: https://mana.mozilla.org/wiki/display/BIDW/MySQL+5.6.17+to+5.6.30+Upgrade

Update MySQL in puppet - in db_mysql/databases.pp or db_mysql/dbbackups.pp change:
From:

```
$mysql_package_type = 'mysql56_33'
```

To:

```
$mysql_package_type = $::fqdn ? { /^developer1/ => 'mysql56_30', default => 'mysql56_33', }
```

42) **Commit the change, and note this revision number (e.g. "Committed revision 117965.")**

```
svn commit -m "Rollback host developer2 Bugs 1292084 and 1302122"
```

43) **Update packages for Kernel and others as the removal of packages from mysql may break yum-wrapper.**

```
Update grub.conf and set previous kernel as default.
vim /boot/grub/grub.conf  -- update default to 1
cat /boot/grub/grub.conf
```

44) **Back on the host, remove the existing MySQL packages:**
In server developer1.db.scl3.mozilla.com:

```
yum-wrapper remove mysql-community-libs-5.6.33-2.el6.x86_64 mysql-community-common-5.6.33-2.el6.x86_64 mysql-community-client-5.6.33-2.el6.x86_64 mysql-community-libs-compat-5.6.33-2.el6.x86_64
mysql-community-server-5.6.33-2.el6.x86_64


OK to remove the following packages, they will be reinstated later (during puppet runs):
 collectd-mysql
 cronie
 cronie-anacron
 crontabs
 nagios-plugins-all
 nagios-plugins-mailq
 nagios-plugins-mysql
 percona-toolkit
 percona-xtrabackup
 perl-DBD-MySQL
 postfix
 Sysstat

REBOOT
```

45) When the machine comes back up
In server developer1.db.scl3.mozilla.com:
Run puppet:

```
/usr/local/sbin/puppetctl run
```

Make sure that the version is equal to or greater than the revision number you noted above (the last "Info" line should be something like):

```
Info: Applying configuration version '117965'
```

Check the status with:

```
/usr/local/sbin/puppetctl status
```

46) Run the puppet again:
In server developer1.db.scl3.mozilla.com:
Run puppet:

```
/usr/local/sbin/puppetctl run
```

47) Edit the /etc/my.cnf to turn of binlogging. Comment log-bin, expire_log_days, log_slave_updates. Check /var/lib/mysql
In server developer1.db.scl3.mozilla.com:
The cnf files need to have bin logs entries off for the mysql_upgrade work.
Comment:

```
log-bin
log_slave_updates
expire_log_days
```

Add:

```
skip-slave-start
```

`/var/lib/mysql`needs to be a symbolic link pointing to `/data/mysql`

```
ln -s /data/mysql /var/lib/mysql
```

48) Start mysql
In server developer1.db.scl3.mozilla.com:

```
service mysqld start
```

on backup hosts:

```
/data/backups/bin/start-all-dbs
```

(check logs, /var/lib/mysql/mysql.err)

49) **Downgrade MySQL**

In server developer1.db.scl3.mozilla.com:

```
1) Update databases.pp as described in step 5, put the version before upgrade (5.6.30).
2) Stop mysql
3) Run puppet, check that it had downgraded package, update my.cnf and turn off binlog again
4) Start mysql
5) run mysql_upgrade
```

50) While this is running, run puppet, and there should be no errors - it should put binlogging back into /etc/my.cnf:

In server developer1.db.scl3.mozilla.com:

```
/usr/local/sbin/puppetctl run
```

51) RESTART MYSQL AND Check replication and status

In server developer1.db.scl3.mozilla.com:

```
service mysqld restart
```

Log into server and run:

```
show slave status\G
```

52) **Check nagios is all OK except db uptime**

(db uptime will go green after 30 minutes, no need to wait for that check to go green)

```
nagios-scl3: status developer2.db.scl3.mozilla.com:MySQL
```

-- FAIL OVER TO WORK IN THE developer1 ---

53) **Set downtime in nagios (step inside Mana)**

Example for IRC chat #db-alerts, here We downtime the server to be worked and the slave to avoid alarm of the replication.

```
nagios-scl3: downtime developer2.db.scl3.mozilla.com 75m upgrade kernel and mysql packages for Bugs1302122  1292084
nagios-scl3: downtime developer1.db.scl3.mozilla.com:MySQL Replication 75m upgrade kernel and mysql packages for Bugs1302122  1292084
```

Reference in : https://mana.mozilla.org/wiki/display/BIDW/How+to+manually+fail+over+a+MySQL+master#HowtomanuallyfailoveraMySQLmaster-Nagiosdowntime

54) **Change Puppet**

Change puppet so the "want_cron" variable in the "checksums" block (and any other block) is "true" on the soon-to-be master, and "false" on the current master. Look for and change all instances of "want_cron" unless there are notes not to.Change puppet so the "want_cron" variable in the "checksums" block (and any other block) is "true" on the soon-to-be master, and "false" on the current master. Look for and change all instances of "want_cron" unless there are notes not to.

To:

```
want_cron =>  $::fqdn ? { /^developer1/ => true, default   => false },
```

Change puppet so the "server_role" variable is "master" on the soon-to-be master, and "slave" on the current master. This is in trunk/manifests/nodes, are in the databases.pp file

To:

```
server_role => $::fqdn ? { /^developer1/ => 'master', default=> 'slave', },


svn commit -m "Upgrading host developer1 Bugs 1292084 and 1302122"
```

Have a Mozilla data team person (Natalie or mpressman) change the nagios checks:

RESUME HERE: Run "/usr/local/sbin/puppetctl run" on both the soon-to-be and current masters (make sure it runs the revision number of your commit)

- Run "/usr/bin/pt-config-diff localhost /etc/my.cnf" on both machines.
  - Puppet changed the /etc/my.cnf.
  - Set read_only to OFF in developer2, and set it to OFF in developer1. AFTER fail over we set to ON in developer1.

55) **Fail over The Load balancer**

Open the following URLS, for the VIPs:

https://10.22.8.209:9090/apps/zxtm/?name=developer-ro-db&section=Virtual%20Servers%3AEdit - Read Only VIP

https://10.22.8.209:9090/apps/zxtm/?name=developer-rw-db&section=Virtual%20Servers%3AEdit - Read Write VIP

Log in and go to the stage-rw-db:

- Move Read Only VIP to developer1.db.scl3.mozilla.com which is 10.22.70.25:3306
- Set the current master (aka 10.22.70.25:3306) to "Draining" -> Save and click "Update"
- On developer2, use "show full processlist;" to see the number of active/sleeping connections for about roughly 15 seconds.
- After about 15 seconds, check the number of remaining connections. If only slave threads, nagios daemon, collectd and newrelic remain, move to the next step. If several to many non-slave threads still exist, stop the mysql service on the master with a normal mysql shutdown.
- In Zeus, set the developer2 (10.22.70.26:3306) to "Disabled" and the developer1 (10.22.70.25:3306) to "Active"
  Set read_only on in mysql instance at developer2 (10.22.70.26:3306).
- Triple check the read_only variable (you should have double checked it above) - it should be ON for developer2 and OFF for developer1.

56) **In developer2.db.scl3.mozilla.com Shutdown Instance**
Shutdown DB and Apply the packages upgrade
Shutdown databases as necessary.

```
service mysqld stop
```

57) **Open Mana page as reference and Update databases.pp file**
Reference: https://mana.mozilla.org/wiki/display/BIDW/MySQL+5.6.17+to+5.6.30+Upgrade

Update MySQL in puppet - in db_mysql/databases.pp:

From:

```
$mysql_package_type = $::fqdn ? { /^developer2/ => 'mysql56_30', default => 'mysql56_33', }
```

To:

```
$mysql_package_type = 'mysql56_30'
```

58) **Commit the change, and note this revision number (e.g. "Committed revision 117965.")**

```
svn commit -m "Rollback host developer1 Bugs 1292084 and 1302122"
```

59) **Update packages for Kernel and others as the removal of packages from mysql may break yum-wrapper.**
In server developer2.db.scl3.mozilla.com:

```
Update grub.conf and set previous kernel as default.
vim /boot/grub/grub.conf  -- update default to 1
cat /boot/grub/grub.conf
```

60) **Back on the host, remove the existing MySQL packages:**
In server developer2.db.scl3.mozilla.com:

```
yum-wrapper remove mysql-community-libs-5.6.33-2.el6.x86_64 mysql-community-common-5.6.33-2.el6.x86_64 mysql-community-client-5.6.33-2.el6.x86_64 mysql-community-libs-compat-5.6.33-2.el6.x86_64
mysql-community-server-5.6.33-2.el6.x86_64


OK to remove the following packages, they will be reinstated later (during puppet runs):
 collectd-mysql
 cronie
 cronie-anacron
 crontabs
 nagios-plugins-all
 nagios-plugins-mailq
 nagios-plugins-mysql
 percona-toolkit
 percona-xtrabackup
 perl-DBD-MySQL
 postfix
 Sysstat
```

REBOOT server

61) When the machine comes back up
In server developer2.db.scl3.mozilla.com:

Run puppet:

```
/usr/local/sbin/puppetctl run
```

Make sure that the version is equal to or greater than the revision number you noted above (the last "Info" line should be something like):

```
Info: Applying configuration version '117965'
```

Check the status with:

```
/usr/local/sbin/puppetctl status
```

62) Run the puppet again:
In server developer2.db.scl3.mozilla.com:
Run puppet:

```
/usr/local/sbin/puppetctl run
```

63) Edit the /etc/my.cnf to turn of binlogging. Comment log-bin, expire_log_days, log_slave_updates. Check /var/lib/mysql
In server developer2.db.scl3.mozilla.com:
The cnf files need to have bin logs entries off for the mysql_upgrade work.
Comment:

```
log-bin
log_slave_updates
expire_log_days
```

Add:

```
skip-slave-start
```

`/var/lib/mysql`needs to be a symbolic link pointing to `/data/mysql`

```
ln -s /data/mysql /var/lib/mysql
```

64) Start mysql
In server developer2.db.scl3.mozilla.com:

```
service mysqld start
```

on backup hosts:

```
/data/backups/bin/start-all-dbs
```

(check logs, /var/lib/mysql/mysql.err)

65) **Downgrade MySQL**
In serverdeveloper2.db.scl3.mozilla.com:

```
1) Update databases.pp as described in step 5, put the version before upgrade.
2) Stop mysql
3) Run puppet, check that it had downgraded package, turn off bin log again
4) Start mysql
5) run mysql_upgrade
```

66) While this is running, run puppet, and there should be no errors - it should put binlogging back into /etc/my.cnf:
In server developer2.db.scl3.mozilla.com:

```
/usr/local/sbin/puppetctl run
```

67) RESTART MYSQL AND Check replication and status
In server developer2.db.scl3.mozilla.com:

```
service mysqld restart
```

Log into server and run:

```
show slave status\G
```

68) Check nagios is all OK except db uptime
(db uptime will go green after 30 minutes, no need to wait for that check to go green)

```
nagios-scl3: status developer2.db.scl3.mozilla.com:MySQL
```

69) **In the spreadsheet, update 'date work' and 'dba' columns**
Take note of the approximate time work was done and update the bug with it. Include hour:minute if possible.

70) **Move the RO vip**
Login to the load balancer at https://www.zlb.ops.scl3.mozilla.com:9090/apps/zxtm/login.cgi with LDAP credentials
Go to stage-ro-db pool at https://10.22.8.209:9090/apps/zxtm/?name=developer-ro-db&section=Virtual%20Servers%3AEdit
Set 10.22.70.25:3306 to “active” and 10.22.70.26 to “disabled”