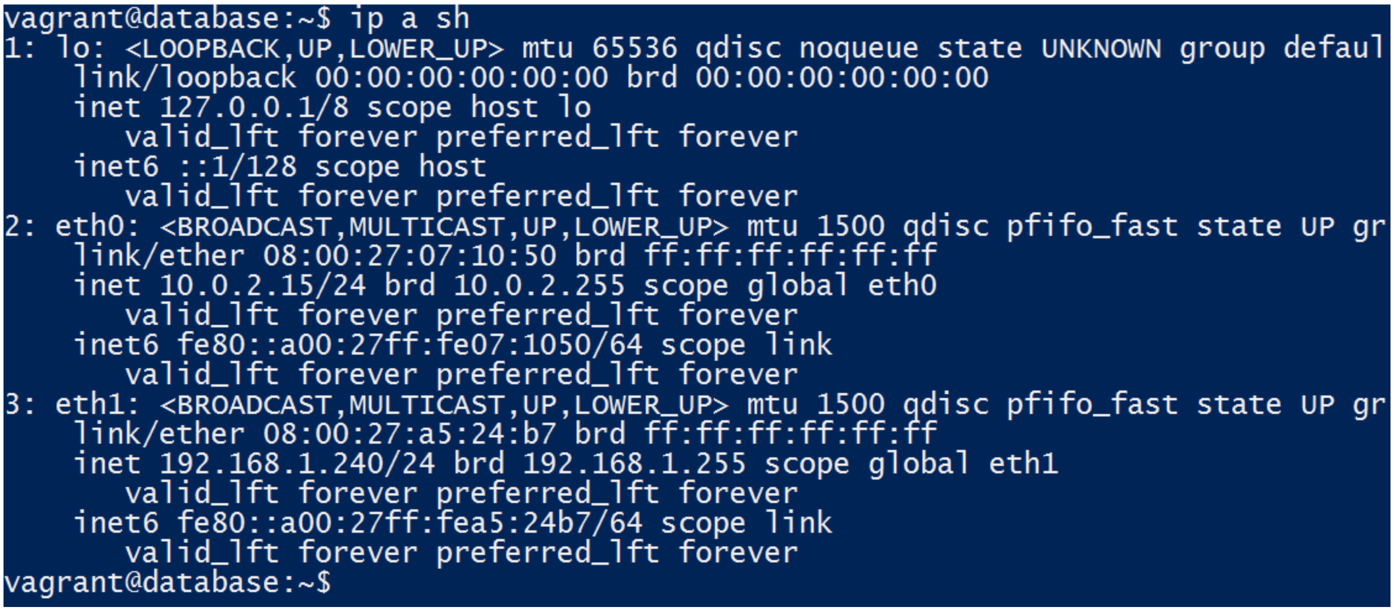
Database - Tutorial

Wednesday, February 8, 2017 3:18 PM

Upon changing the public network value in your Vagrantfile and issuing a vagrant up and vagrant ssh command, you should be able to issue: `ip a sh` command and see similar output: Note if you are using an OS that uses systemd you will not see eth0 or eth1 but enp0sX style bus enumeration of devices.



```
vagrant@database:~$ ip a sh
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gr
    link/ether 08:00:27:07:10:50 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe07:1050/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gr
    link/ether 08:00:27:a5:24:b7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.240/24 brd 192.168.1.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fea5:24b7/64 scope link
       valid_lft forever preferred_lft forever
vagrant@database:~$
```

Screen clipping taken: 2/9/2017 12:35 PM

The key is to see the 2nd device with your public IP.

Once successful issue these commands to retrieve the sample code needed:

1. `git clone https://github.com/jhajek/db-samples`
2. `cd db-samples`
3. `chmod +x *.sh`
4. `./mysql-server-install.sh`
   a. (note that it prompts you for your password the way to automate that out of the way is here: **but this is not required at this time:**)- the way to seed that is located here:
   b. http://stackoverflow.com/questions/7739645/install-mysql-on-ubuntu-without-password-prompt)

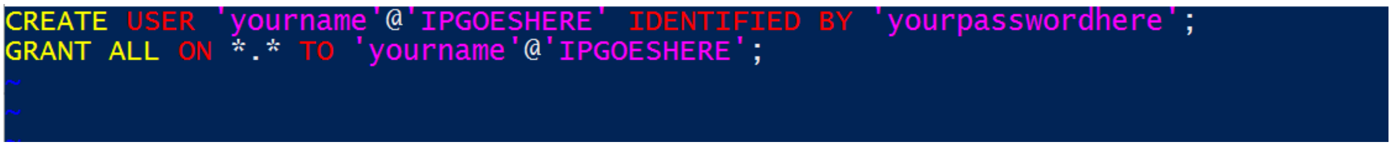
```
sudo debconf-set-selections <<< 'mysql-server mysql-server/root_password
password your_password'
sudo debconf-set-selections <<< 'mysql-server mysql-
server/root_password_again password your_password'
sudo apt-get -y install mysql-server
```

From <http://stackoverflow.com/questions/7739645/install-mysql-on-ubuntu-without-password-prompt>

Upon completion of the above steps you need to execute 3 `.sql` script files to correctly setup the infrastructure needed for the database server.

1. `create.sql` - which will create and build your database and table
2. `insert.sql` - which will insert 3 records of data
3. `create-users-with-grants.sql` - which will create you a non-root user (always a good idea) and grant them permission to connect. This script is wide-open permission wise and needs

to be reduced - note that you need to replace IPGOESHERE with the public IP of the webserver vagrant box.



```
CREATE USER 'yourname'@'IPGOESHERE' IDENTIFIED BY 'yourpasswordhere';
GRANT ALL ON *.* TO 'yourname'@'IPGOESHERE';
```

Screen clipping taken: 2/9/2017 3:56 PM

Change `'yourname'@'IPGOESHERE' IDENTIFIED BY 'yourpasswordhere';` to: `'worker'@'192.168.1.239' IDENTIFIED BY 'ilovebunnies';`

The name worker is arbitrary but should be something that by its definition tells you that this is a limited ability user (by design). IDENITIFED BY is the password for this user.

The next line: `GRANT ALL ON *.* TO 'yourname'@'IPGOESHERE';` change this to: `GRANT ALL ON *.* TO 'worker'@'192.168.1.239';`

The GRANT ALL gives this worker ALL permissions (probably not a good idea, you should limit the power by specifically granting privileges (the less the better!))
https://dev.mysql.com/doc/refman/5.7/en/grant.html

The second portion: `ON *.*` grants access to all databases and all tables (again not secure by design -- should be `store.items` in this case. The pattern is `dbname.tablename` * is a wildcard.

The third portion: `TO 'worker'@'192.168.1.239'` is where you grant access to a remote IP, 192.168.1.239 is my webserver (replace it with your webserver IP). You can also use a % as a wildcard in this case.

You can execute them from the command line:

```
mysql -u root -p < create.sql
```

The commandline redirection will execute the sql file into the database. There is a way to pre-seed your my.cnf file with a username and password so you do not have to enter a password on the command line manually **but at this point this is not requried:**

Best thing to do is create a `~/.my.cnf`
https://easyengine.io/tutorials/mysql/mycnf-preference/

Add these entries -- this allows for customization without having to edit the /etc/mysql/my.cnf

```
[client]
user=mysqluser
password=mysqlpass
```

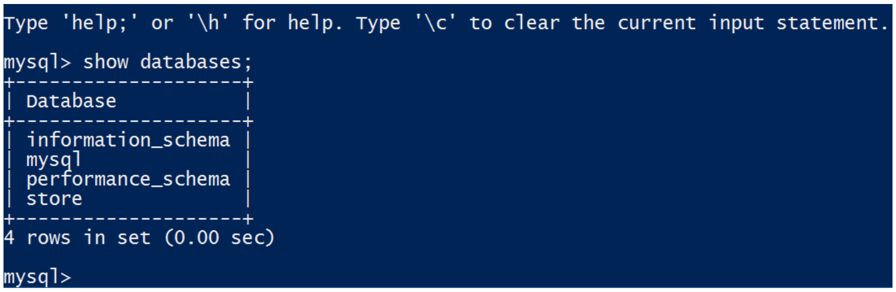
From <https://easyengine.io/tutorials/mysql/mycnf-preference/>

Upon successful completion of the above steps you will be able to log into the mysql database from the command line (as root, your user you created won't have the proper CONNECT permission--by design).

```
mysql -u root -p
```

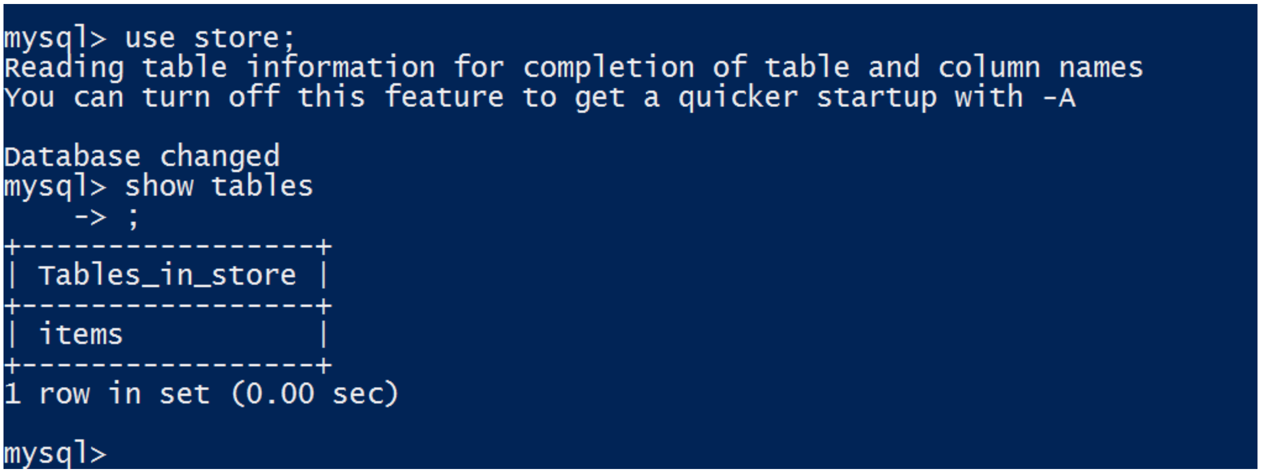
Upon successful login at the mysql prompt:

```
show databases;
```



```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| store              |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

You should see the database named **store** has been created
Next step is issue the command: `use store;`

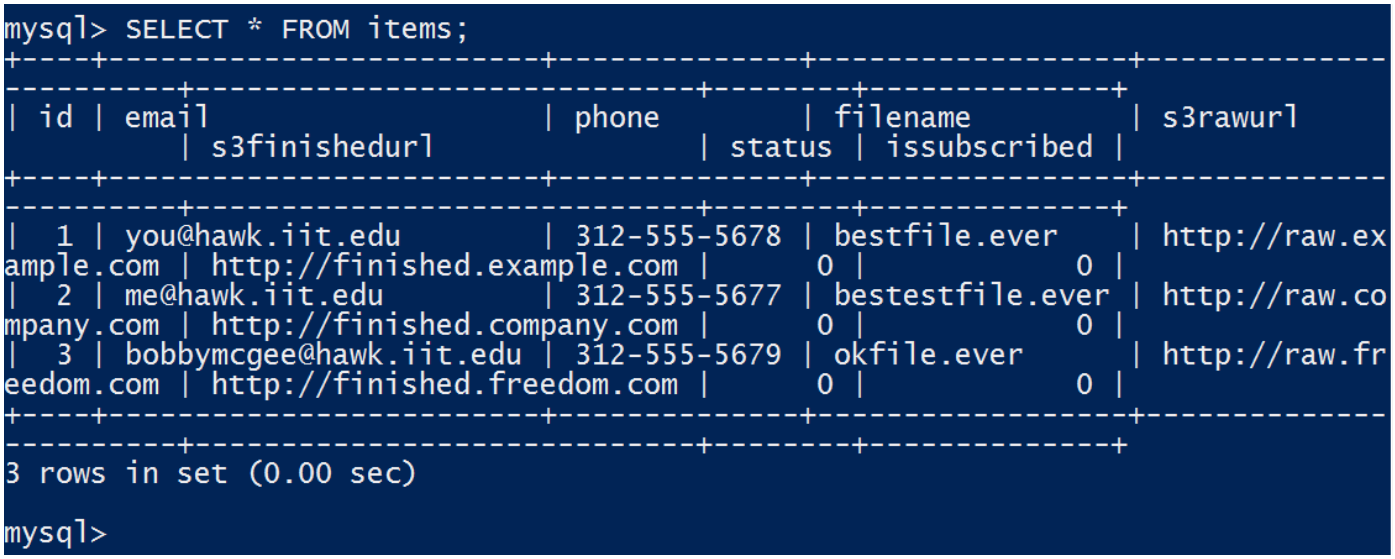


```
mysql> use store;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables
    -> ;
+-----------------+
| Tables_in_store |
+-----------------+
| items           |
+-----------------+
1 row in set (0.00 sec)

mysql>
```

Screen clipping taken: 2/9/2017 12:30 PM

Next step issue the command `show tables;` You should see the above output.

Final step to make sure the **insert.sql** script worked properly is to issue: `SELECT * FROM items;`

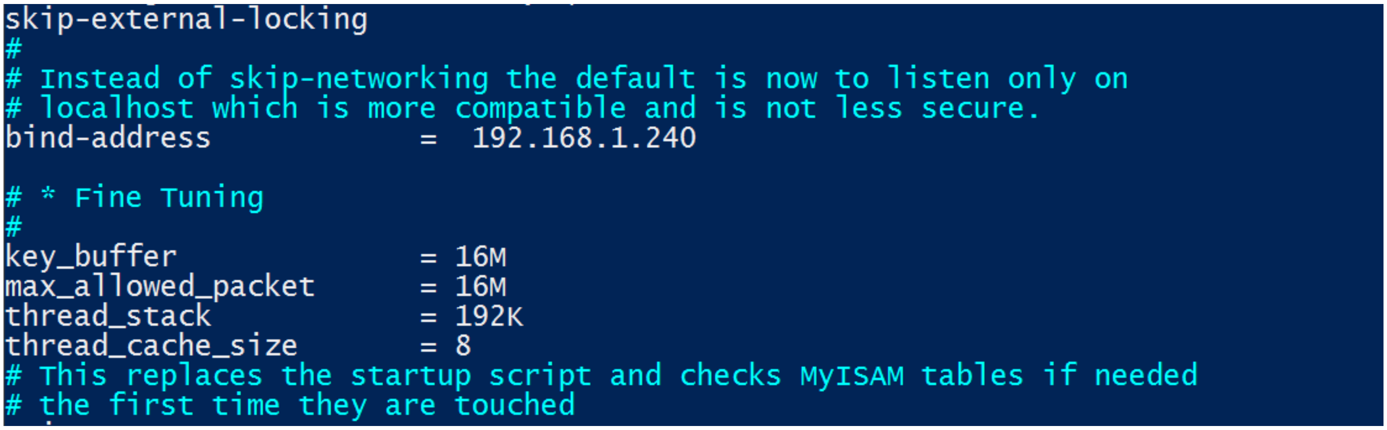


```
mysql> SELECT * FROM items;
+----+-------------------------+--------------+------------------+-------------
----------+-------------------------------+--------+--------------+
| id | email                   | phone        | filename         | s3rawurl
          | s3finishedurl                 | status | issubscribed |
+----+-------------------------+--------------+------------------+-------------
----------+-------------------------------+--------+--------------+
|  1 | you@hawk.iit.edu        | 312-555-5678 | bestfile.ever    | http://raw.ex
ample.com | http://finished.example.com |      0 |            0 |
|  2 | me@hawk.iit.edu         | 312-555-5677 | bestestfile.ever | http://raw.co
mpany.com | http://finished.company.com |      0 |            0 |
|  3 | bobbymcgee@hawk.iit.edu | 312-555-5679 | okfile.ever      | http://raw.fr
eedom.com | http://finished.freedom.com |      0 |            0 |
+----+-------------------------+--------------+------------------+-------------
----------+-------------------------------+--------+--------------+
3 rows in set (0.00 sec)

mysql>
```

Screen clipping taken: 2/9/2017 12:32 PM

The output wraps but it should look like the screen above:

The final step is to modify your mysql configuration file. It is located at: `/etc/mysql/my.cnf`



```
skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            =  192.168.1.240

# * Fine Tuning
#
key_buffer              = 16M
max_allowed_packet      = 16M
thread_stack            = 192K
thread_cache_size       = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
```

Screen clipping taken: 2/9/2017 3:28 PM

You need to modify the value: `bind-address` to be the public IP of the database server (the server this is running on, not the IP of a server connection to it--warning this opens the mysql server to public connections - you should limit these via mysql GRANT privileges or your firewall/iptables).

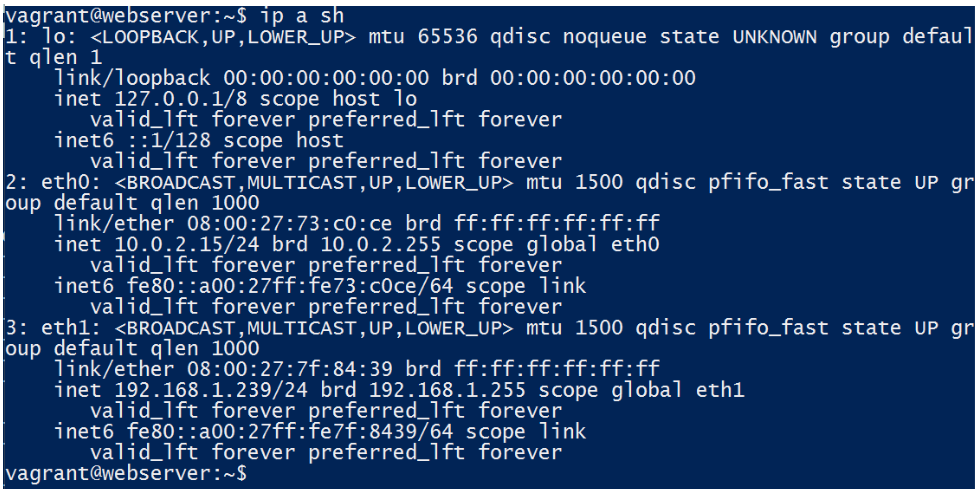
Save the my.cnf file and then reload the mysql service: `sudo service mysql reload`

If you have achieved this step the your database Vagrant box is setup and complete. No further action is required.

Webserver - Tutorial

Thursday, February 9, 2017 3:09 PM

Upon changing the public network value in your Vagrantfile and issuing a vagrant up and vagrant ssh command, you should be able to issue: `ip a sh` command and see similar output: Note if you are using an OS that uses systemd you will not see eth0 or eth1 but enp0sX style bus enumeration of devices.



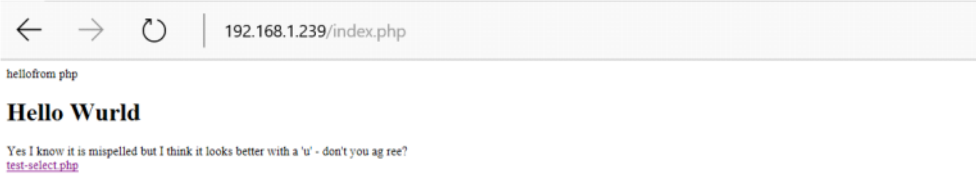
```
vagrant@webserver:~$ ip a sh
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gr
oup default qlen 1000
    link/ether 08:00:27:73:c0:ce brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe73:c0ce/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gr
oup default qlen 1000
    link/ether 08:00:27:7f:84:39 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.239/24 brd 192.168.1.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe7f:8439/64 scope link
       valid_lft forever preferred_lft forever
vagrant@webserver:~$
```

Screen clipping taken: 2/9/2017 3:12 PM

The key is to see the 2nd device with your public IP.

Once successful issue these commands to retrieve the sample code needed:

1. `git clone https://github.com/jhajek/db-samples`
2. `cd db-samples`
3. `chmod +x *.sh`
4. `./mysql-client-install.sh`
5. `sudo cp *.php /var/www/html` (assuming your pwd is `~/db-samples`)
6. Open a webbrowser on your HOST system (your Windows or Mac or Linux system) and enter in the public IP of your webserver. In the screenshot above, mine would be 192.168.1.239.
    a. http://192.168.1.239/index.php



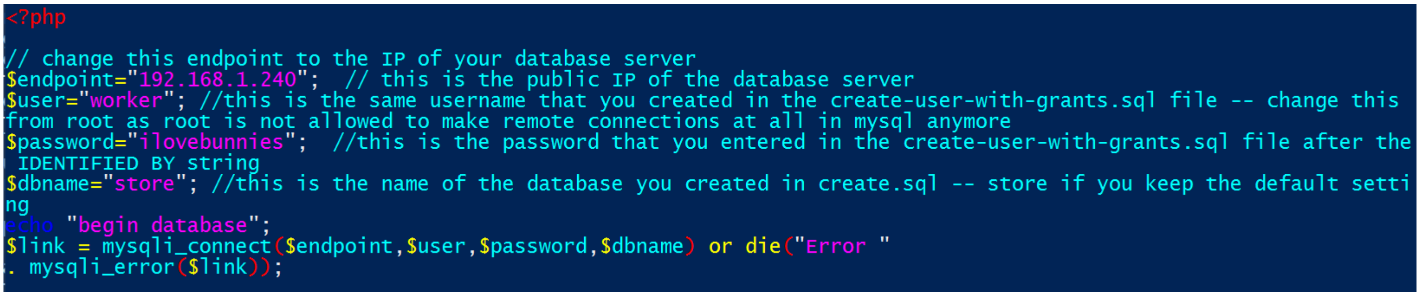
hellofrom php

Hello Wurld

Yes I know it is mispelled but I think it looks better with a 'u' - don't you ag ree?
test-select.php

Screen clipping taken: 2/9/2017 3:55 PM

    b. You should see a "Hello World message" (with a small joke,) with a link to `test-select.php`
7. Open this file in **vim** or **nano (**test-select.php**)**
    a. Edit the 4 variables at the top of the program with the corresponding values

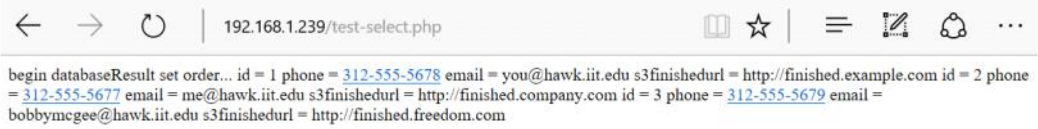


```
<?php

// change this endpoint to the IP of your database server
$endpoint="192.168.1.240";  // this is the public IP of the database server
$user="worker"; //this is the same username that you created in the create-user-with-grants.sql file -- change this
from root as root is not allowed to make remote connections at all in mysql anymore
$password="ilovebunnies";  //this is the password that you entered in the create-user-with-grants.sql file after the
 IDENTIFIED BY string
$dbname="store"; //this is the name of the database you created in create.sql -- store if you keep the default setti
ng
echo "begin database";
$link = mysqli_connect($endpoint,$user,$password,$dbname) or die("Error "
. mysqli_error($link));
```

Screen clipping taken: 2/9/2017 4:19 PM

Click the link in index.php to the test-select.php file.

If successful you will see this output:



```
begin databaseResult set order... id = 1 phone = 312-555-5678 email = you@hawk.iit.edu s3finishedurl = http://finished.example.com id = 2 phone
= 312-555-5677 email = me@hawk.iit.edu s3finishedurl = http://finished.company.com id = 3 phone = 312-555-5679 email =
bobbymcgee@hawk.iit.edu s3finishedurl = http://finished.freedom.com
```

Screen clipping taken: 2/9/2017 4:16 PM

If you receive and error message the place to start checking on your webserver vagrant box is:
`sudo tail /var/log/apache2/error.log`, which is the location of the Apache2 error logs. There are many clues here and it should be the first place you check when an error occurs.