Conversation with:
Ali Hassan
Database Developer

Database Administrator Adv. Challenge Chat Transcript

Ali sent:
Hey @playerone, nice to meet you! I'm told you are the new Database Administrator intern.

Welcome to the TryCyber Bathroom Depot tech division 🚽! I'm Ali Hassan, one of the Database Developers that work here, and I'll be your mentor for today.

Ali sent:
I'm going to be showing you how to perform some of the common tasks of a **Database Administrator**. I usually perform my tasks on a server like the **Ubuntu 22.04 Linux** server you are looking at now.

Ali sent:
All the tasks you'll be performing today will require you to use the **Terminal** and a text editor. If you're unfamiliar with using a terminal, that's fine! I can bring you up to speed with a brief tutorial on it.

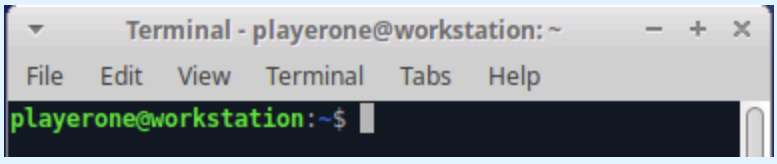
Ali sent:
Shall we get started? 🙄

Participant sent:
Terminal tutorial, please!

Ali sent:
The **Terminal** is primarily used to access an application called a shell. We interact with shells via a **command-line interface (CLI)** where we input and execute text-based commands on a system.

Ali sent:
Let's open the Terminal on this system now and run a few basic commands so you get the hang of it. You can open the **Terminal** by double-clicking the icon on the desktop named **Terminal Emulator**. It should look like this once you have it open...

Ali sent:



Participant sent:
I've opened the Terminal.

Ali sent:
Great. Basically, you type commands into the shell within the Terminal and then hit `Enter` or `Return` on the keyboard to run the command.

Ali sent:
To properly utilize any CLI command, you must understand its format, options, and other arguments. The **format** defines the structure and order of options and other arguments, the **options** define or modify behavior, and the **arguments** are command defined input types (e.g., files, directories, users, software packages, etc.).

Ali sent:
Additionally, all CLI commands on this system are case-sensitive, so pay close attention to the capitalization of letters.

Ali sent:
The basic command format is...

```
command [OPTIONS] [ARGUMENTS]
```

However, it varies a lot from command to command, and options and other arguments are not always required.

Ali sent:
An example of a command that does not require any options or other arguments is `whoami`. If you type `whoami` into the shell and then hit `Enter` or `Return` on the keyboard, the `whoami` command will output the name of the user running the command into the Terminal. It should look like this if you run that command...

Ali sent:

```
playerone@workstation:~$ whoami
playerone
```

Participant sent:
What's a more involved example?

Ali sent:
I'll give you a more complex example using the `ls` command.

Ali sent:
In this example, we will use the `ls` command to list detailed information about the entire contents of the `Templates` directory (i.e., folder) in your (playerone's) `home` directory. Our example `ls` command will use the following format...

```
ls [OPTIONS] [DIRECTORY]
```

Note that in this case, the command's argument requires the input to be a directory.

Ali sent:
The actual command we want to run is...

```
ls -a -l /home/playerone/Templates
```

Ali sent:
The `-a` and the `-l` are both options that modify the `ls` command's behavior, and `/home/playerone/Templates` is our argument which is a directory provided in the form of a **path**.

Ali sent:
For this command's argument, we must provide the **path** to the directory from the root of the file system; otherwise, the system won't know which directory named `Templates` we are referring to.

Ali sent:
And finally, when you run that command, the output should look like this...

Ali sent:
```
playerone@workstation:~$ ls -a -l /home/playerone/Templates/
total 24
drwxr-xr-x  2 playerone playerone 4096 Jul 12 21:36  .
drwxr-x--- 14 playerone playerone 4096 Jul 12 21:36  ..
-rw-r--r--  1 playerone playerone 6859 Apr  9  2022 'OpenDocument Spreadsheet.ods'
-rw-r--r--  1 playerone playerone 7388 Apr  9  2022 'OpenDocument Text.odt'
-rw-r--r--  1 playerone playerone    0 Apr  9  2022 'Plain Text.txt'
```

Participant sent:
Got it. Any extra notes?

Ali sent:
The Terminal and shells are incredibly powerful and versatile tools. Not all commands and programs follow the general structure I've provided you with here. Unfortunately, we only have time to cover the basics, but I do have a few last things I'd like to mention.

Ali sent:
Some commands will not print visual output to the Terminal in normal operation, such as the `cp` command, which is used to copy files and directories.

Ali sent:
Many commands can use multiple arguments, handle multiple input types for arguments (e.g., file and/or directory paths), or have options that will have their own arguments.

Ali sent:
Some options can, or must be, written in a long-form format (e.g., `ls --all [DIRECTORY]` is the same as `ls -a [DIRECTORY]`).

Ali sent:
Options without arguments can often be provided together and in any order (e.g., `ls -la [DIRECTORY]` is the same as `ls -a -l [DIRECTORY]`).

Ali sent:
And last, but certainly not least, you can almost always reference a command's format, options, and other arguments using the command `man [COMMAND]` (e.g., `man cp`) to view the provided command's manual page in the Terminal.

Ali sent:
Hopefully that was not too much information! I know it seems like a lot, but it gets easier the more you use it. For today's tasks, I'll be sure to provide you with more details for any commands and programs you'll need.

Participant sent:
Sounds good! I'm ready to get started!

Ali sent:
Excellent. Today, we'll be working with a **MySQL database**. As some quick background information, MySQL is an open-source relational database management system, and a pretty popular one at that.

Ali sent:
We are planning on using a MySQL database as part of a new application our division is developing. We want to use the MySQL database to store a variety of data for that application, including things like employee names and the types of products TryCyber Bathroom Depot offers. 🛁

Ali sent:
Today, I want your help with two tasks, both of which involve reconfiguring settings on the soon-to-be development MySQL database on the Ubuntu Linux server in front of you. 🔧

Ali sent:
Configuring and reconfiguring databases are common responsibilities for Database Administrators.

Participant sent:
Good to know! Where do we start?

Ali sent:
The first task I'd like you to take care of will be **making MySQL accessible over the network**.

Ali sent:
Our application will be on a different server on the same network, which is why we need to make the database available over the network.

Ali sent:
To do that, you will need to reconfigure the MySQL `bind_address` **setting**.

Ali sent:
But, before I give you more details on how to do that, let's see what the `bind_address` setting is currently set to in the running MySQL configuration. To do that, run the following command in the Terminal...

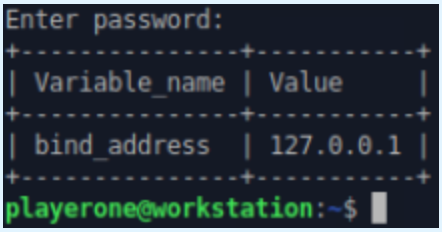
```
mysql -u playerone -p -e 'show variables like "bind_address";'
```

When you run this command, you will be prompted for your password. Just enter your password into the prompt and hit `Enter` or `Return` on the keyboard when you're done. While typing your password into this prompt, you will NOT see any indication that you are typing in the Terminal; this is normal behavior. (Note: Your password can be found on the Info Tab)

Ali sent:
That `mysql` command uses the `-e` option to execute the SQL statement in single quotes, `show variables like "bind_address";`, on the MySQL server. The `-u` and `-p` options are for authenticating to the MySQL server, `-u` is for providing the user, and `-p`, without an argument, requests a password prompt.

Ali sent:
The output of that command should look like this...

Ali sent:



Participant sent:
Got it.

Ali sent:
As expected, and as shown in that output, the `bind_address` setting in the running MySQL configuration is set to `127.0.0.1`, which is a special network address known as the loopback address. This is the default configuration setting set during MySQL installation and essentially makes MySQL only locally available on this server.

Ali sent:
Alright, let's get to reconfiguring that setting!

Ali sent:
I'll want you to make MySQL network accessible by updating the `bind-address` **setting** in the `mysqld.cnf` **configuration file** (found at the path `/etc/mysql/mysql.conf.d/mysqld.cnf`) and then **restarting** the `mysql` **service**.

Ali sent:
As a quick note, because it can be a little confusing, `bind-address` in `mysqld.cnf` is the same setting as `bind_address` in the MySQL running configuration.

Ali sent:
Okay, to edit the `mysqld.cnf` file, we need to open it in a text editor with elevated permissions. To do that, run the following command in the Terminal...

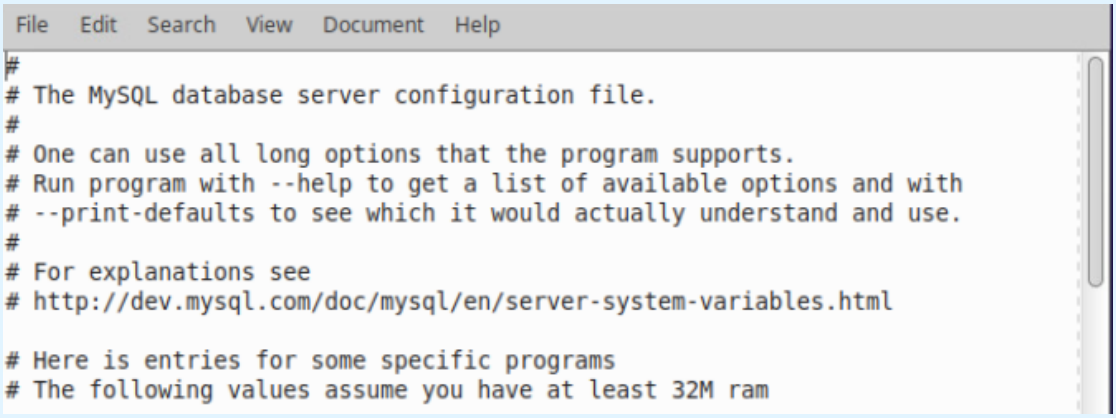
```
sudoedit /etc/mysql/mysql.conf.d/mysqld.cnf
```

That command will open the `mysqld.cnf` file at the provided path, with the default text editor (`mousepad` in this case) running as the superuser.

Ali sent:
When you use the `sudoedit` command, you will sometimes be prompted for your password. Just enter your password into the prompt and hit `Enter` or `Return` on the keyboard when you're done. (Note: Your password can be found on the Info Tab)

Ali sent:
You should now be looking at a graphical text editor, with a file open that looks like this...

Ali sent:



Participant sent:
I'm editing that file now.

Ali sent:
Great! You're now looking at the MySQL server configuration file, `mysqld.cnf`. This is where we will update the `bind-address` setting to `0.0.0.0`, which will configure MySQL to listen on all available network interfaces and make it available on the network. Just so you know, this is NOT the value for the `bind-address` setting we would use in production, nor will we leave this development server this way for long. We will narrow this down to specific network addresses or ranges at a later date.

Ali sent:
The line in this configuration file that contains the `bind-address` configuration setting is line 31. You can scroll down until you find it, or you could use the Find feature in the `mousepad` text editor under the Search section of the menu bar and search for what the line contains right now (i.e., 'bind-address').

Ali sent:
Once you find the right line in the configuration file, you need to update the line from reading...

```
bind-address = 127.0.0.1
```

to read...

```
bind-address = 0.0.0.0
```

Ali sent:
Which would look like this...

Ali sent:

```
bind-address            = 0.0.0.0
```

Participant sent:
I've updated the line.

Ali sent:
Great!

Ali sent:
Now, if you have not already, you will need to save the file and close the text editor; if you didn't save it before trying to close it, it will ask you if you want to save the changes. As a note, when using `sudoedit`, you must save and close the text editor before the changes are written to the open file.

Ali sent:
With the `bind-address` setting updated in the `mysqld.cnf` configuration file, we now need to restart the `mysql` service to update the running configuration. To restart the `mysql` service, run the following command in the Terminal...

```
sudo systemctl restart mysql
```

The above command utilizes the `sudo` command, which will sometimes prompt you for your password. Just enter your password into the prompt and hit `Enter` or `Return` on the keyboard when you're done. (Note: Your password can be found on the Info Tab)

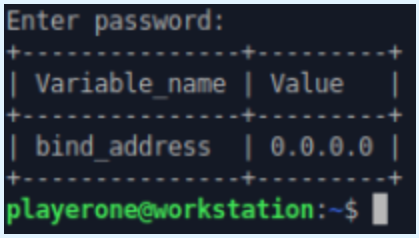
Ali sent:
If you've done everything correctly, that command should not have any output. If you see any errors, you'll likely need to re-open the `mysqld.cnf` file and review the changes you've made to make sure they are exactly as specified.

Ali sent:
To confirm your change to the `mysqld.cnf` file is now in the running configuration, run the following command in the Terminal again...

```
mysql -u playerone -p -e 'show variables like "bind_address";'
```

Ali sent:
This time, the output should look like this...

Ali sent:



Ali sent:
Let me know when you are done with that so we can move on to the last task of the day.

Participant sent:
I'm done.

Ali sent:
Nice!

Ali sent:
The last task I'd like you to take care of will be reconfiguring MySQL to accept only **100 simultaneous connections**. Currently, MySQL is still set to the default amount, which is 151 simultaneous connections. The application developers have asked that we limit it to 100 for now.

Ali sent:
To do that, you will need to reconfigure the MySQL `max_connections` **setting**. Lucky for you, the process of updating the `max_connections` setting is nearly identical to the process we just used to update `bind_address`.

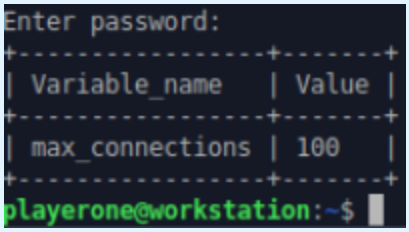
Ali sent:
You will need to edit the `mysqld.cnf` configuration file again, and then once you are done, you will need to restart the `mysql` service again (using the command `sudo systemctl restart mysql`). However, this time you will be updating line 46 of `mysqld.cnf` and updating the value for `max_connections`.

Ali sent:
Once you've correctly updated the file and restarted the `mysql` service, you can verify your change is in the running configuration by running the command...

```
mysql -u playerone -p -e 'show variables like "max_connections";'
```

Ali sent:
The output should look like this...

Ali sent:



Ali sent:
And now it looks like I have a meeting I need to get to. I'm confident you'll be able to handle this last task on your own. Thanks again for all your help so far! 💯