```
}
```

Tutorial: Changing Drive Permissions and Ownership with a Bash Script

This tutorial explains how to create and use a bash script to change the ownership and permissions of files and directories on specified drives. The script automates the process, ensuring all files and directories on the drive have the correct permissions and ownership.

Step-by-Step Explanation

1. Shebang

```
#!/bin/bash
```

The shebang (`#!/bin/bash`) at the top of the script specifies that the script should be run using the Bash shell.

2. Defining Mount Points

```
DRIVES=(
    "/mnt/drives/drive1"
    "/mnt/drives/drive2"
)
```

The `DRIVES` array contains the mount points of the drives whose permissions and ownership need to be changed. Add more drives by adding additional entries.

3. Defining the Username

```
USER="azazelthegray"
```

The `USER` variable specifies the username that will own all files and directories on the drives.

4. Changing Ownership Function

```
change_ownership() {
    local drive="$1"
    echo "Changing ownership of all files and directories in $drive to $USER"
    sudo chown -v -R $USER:$USER "$drive"
}
```

This function changes the ownership of all files and directories on the specified drive to the specified user.

- `local drive="$1"`: Captures the drive path passed to the function.
- `sudo chown -v -R $USER:$USER "$drive"`: Uses chown to recursively (`-R`) change the owner and group of all files and directories on the drive to the specified user. The `-v` flag provides verbose output, showing the changes being made.

5. Changing Permissions Function

```
change_permissions() {
    local drive="$1"
    echo "Changing permissions of all files in $drive"
    sudo find "$drive" -type f -exec chmod -v 644 {} +

    echo "Changing permissions of all directories in $drive"
    sudo find "$drive" -type d -exec chmod -v 755 {} +
}
```

This function changes the permissions of all files and directories on the specified drive.

- `sudo find "$drive" -type f -exec chmod -v 644 {} +`: Uses `find` to locate all files (`-type f`) on the drive and `chmod` to set their permissions to `644` (read and write for owner, read-only for others). The `-v` flag provides verbose output.
- `sudo find "$drive" -type d -exec chmod -v 755 {} +`: Uses `find` to locate all directories (`-type d`) on the drive and `chmod` to set their permissions to `755` (read, write, and execute for owner, read and execute for others).

6. Applying Changes

```
for DRIVE in "${DRIVES[@]}"; do
    change_ownership "$DRIVE"
    change_permissions "$DRIVE"
done

echo "Permissions and ownership have been set for all specified drives."
```

This loop iterates over each drive in the `DRIVES` array and applies the ownership and permissions changes by calling the `change_ownership` and `change_permissions` functions.

Why Use Specific Commands?

`chown`

- `chown` changes the ownership of files and directories. This is necessary to ensure that the specified user has control over the files and directories on the drive.

`chmod`

- `chmod` changes the permissions of files and directories. Setting appropriate permissions ensures that files and directories are accessible only to the intended users.

`find`

- `find` is used to locate files (`-type f`) and directories (`-type d`) on the drive. Using `find` with `-exec` allows us to apply `chmod` to each found file or directory individually, ensuring comprehensive permission changes.

The Script

```bash
#!/bin/bash

# Define the mount points of the drives
DRIVES=(
    "/mnt/drives/drive1"
    "/mnt/drives/drive2"
)

# Define the username for ownership
USER="azazelthegray"

# Function to change ownership of the entire drive
change_ownership() {
    local drive="$1"
    echo "Changing ownership of all files and directories in $drive to $USER"
    sudo chown -v -R $USER:$USER "$drive"
}

# Function to change permissions of all files and directories
change_permissions() {
    local drive="$1"
    echo "Changing permissions of all files in $drive"
    sudo find "$drive" -type f -exec chmod -v 644 {} +

    echo "Changing permissions of all directories in $drive"
    sudo find "$drive" -type d -exec chmod -v 755 {} +
}

# Loop through each drive and apply ownership and permissions changes
for DRIVE in "${DRIVES[@]}"; do
    change_ownership "$DRIVE"
    change_permissions "$DRIVE"
done

echo "Permissions and ownership have been set for all specified drives."
```

Conclusion

This script automates the process of changing ownership and permissions on specified drives, ensuring the correct user owns the files and directories and that permissions are appropriately set for security and accessibility. By running this script, you can maintain a consistent and secure file system across your drives.