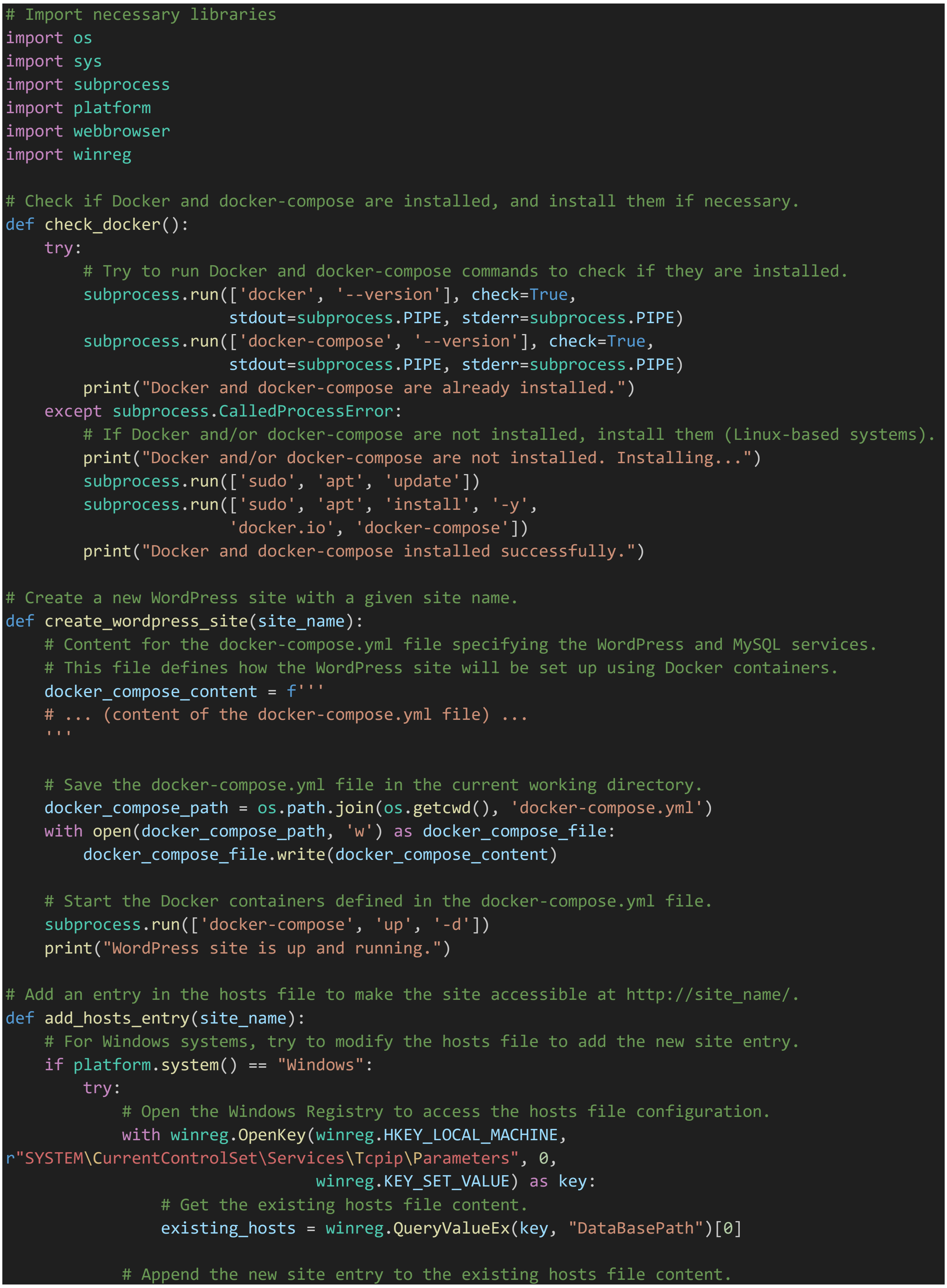
DEVOPS TASK

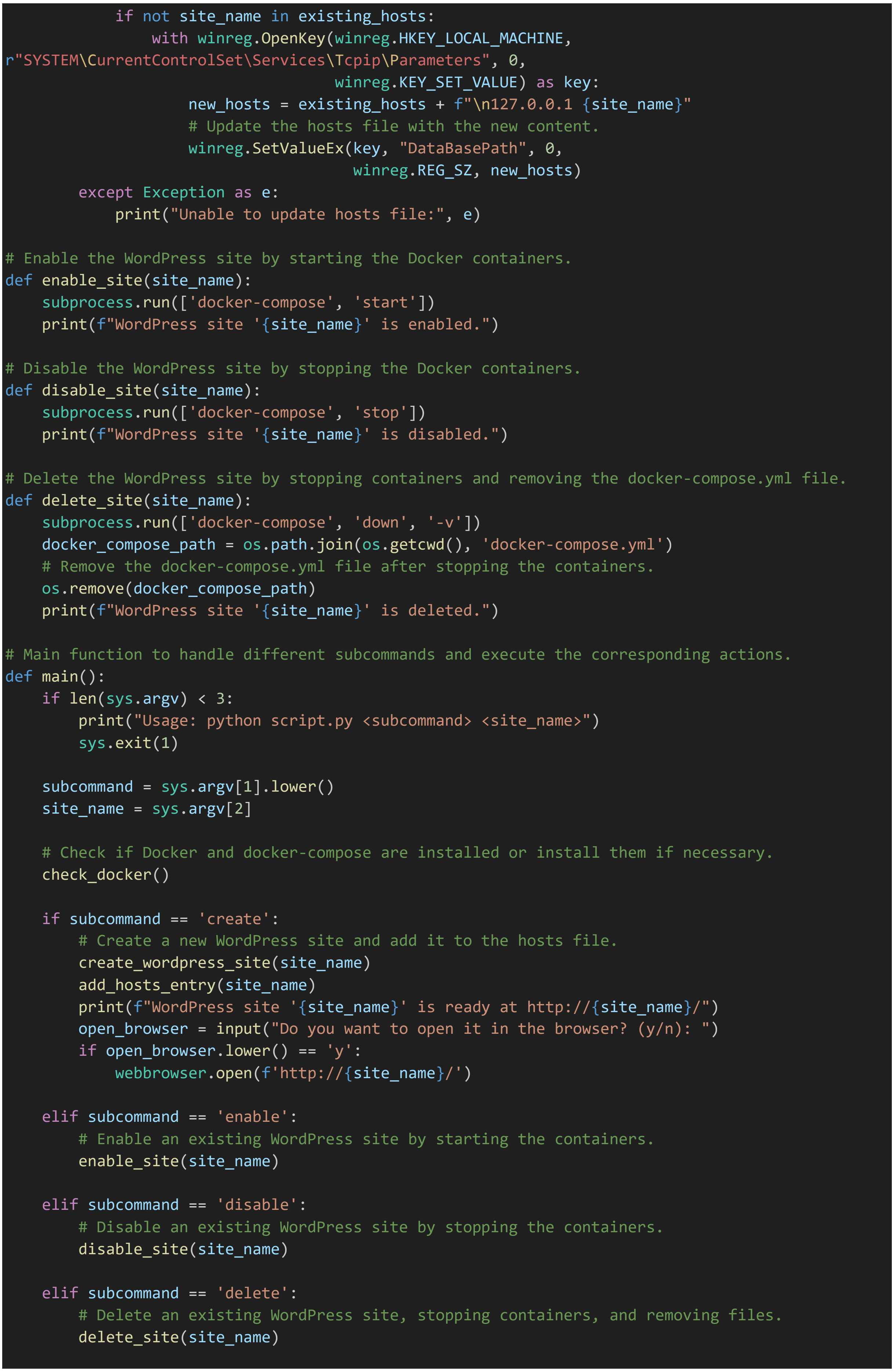
Create a command-line script, preferably in Bash, PHP, Node, or Python to perform the following tasks:

1. Check if `docker` and `docker-compose` is installed on the system. If not present, install the missing packages.
2. The script should be able to create a WordPress site using the latest WordPress Version. Please provide a way for the user to provide the site name as a command-line argument.
3. It must be a LEMP stack running inside containers (Docker) and a docker-compose file is a must.
4. Create a `/etc/hosts` entry for *example.com* pointing to localhost. Here we are assuming the user has provided *example.com* as the site name.
5. Prompt the user to open *example.com* in a browser if all goes well and the site is up and healthy.
6. Add one more subcommand to delete the site (deleting containers and local files).

Script:



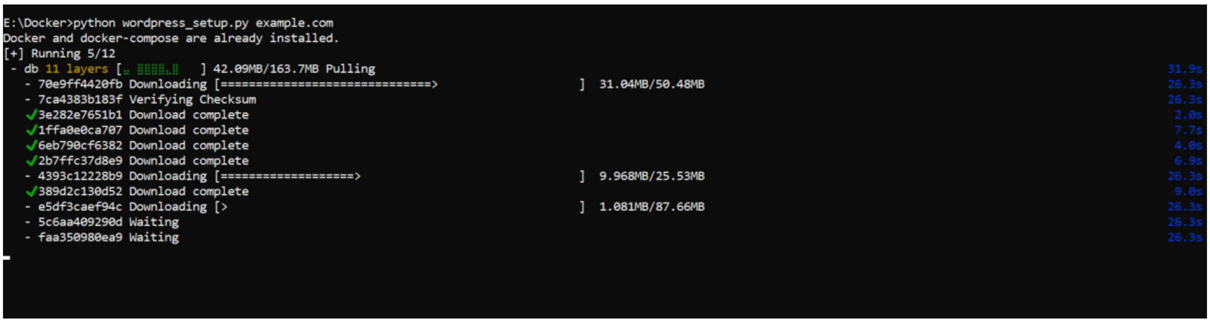
```
# Import necessary libraries
import os
import sys
import subprocess
import platform
import webbrowser
import winreg

# Check if Docker and docker-compose are installed, and install them if necessary.
def check_docker():
    try:
        # Try to run Docker and docker-compose commands to check if they are installed.
        subprocess.run(['docker', '--version'], check=True,
                       stdout=subprocess.PIPE, stderr=subprocess.PIPE)
        subprocess.run(['docker-compose', '--version'], check=True,
                       stdout=subprocess.PIPE, stderr=subprocess.PIPE)
        print("Docker and docker-compose are already installed.")
    except subprocess.CalledProcessError:
        # If Docker and/or docker-compose are not installed, install them (Linux-based systems).
        print("Docker and/or docker-compose are not installed. Installing...")
        subprocess.run(['sudo', 'apt', 'update'])
        subprocess.run(['sudo', 'apt', 'install', '-y',
                       'docker.io', 'docker-compose'])
        print("Docker and docker-compose installed successfully.")

# Create a new WordPress site with a given site name.
def create_wordpress_site(site_name):
    # Content for the docker-compose.yml file specifying the WordPress and MySQL services.
    # This file defines how the WordPress site will be set up using Docker containers.
    docker_compose_content = f'''
    # ... (content of the docker-compose.yml file) ...
    '''

    # Save the docker-compose.yml file in the current working directory.
    docker_compose_path = os.path.join(os.getcwd(), 'docker-compose.yml')
    with open(docker_compose_path, 'w') as docker_compose_file:
        docker_compose_file.write(docker_compose_content)

    # Start the Docker containers defined in the docker-compose.yml file.
    subprocess.run(['docker-compose', 'up', '-d'])
    print("WordPress site is up and running.")

# Add an entry in the hosts file to make the site accessible at http://site_name/.
def add_hosts_entry(site_name):
    # For Windows systems, try to modify the hosts file to add the new site entry.
    if platform.system() == "Windows":
        try:
            # Open the Windows Registry to access the hosts file configuration.
            with winreg.OpenKey(winreg.HKEY_LOCAL_MACHINE,
r"SYSTEM\CurrentControlSet\Services\Tcpip\Parameters", 0,
                                winreg.KEY_SET_VALUE) as key:
                # Get the existing hosts file content.
                existing_hosts = winreg.QueryValueEx(key, "DataBasePath")[0]

            # Append the new site entry to the existing hosts file content.
```



```
            if not site_name in existing_hosts:
                with winreg.OpenKey(winreg.HKEY_LOCAL_MACHINE,
r"SYSTEM\CurrentControlSet\Services\Tcpip\Parameters", 0,
                                    winreg.KEY_SET_VALUE) as key:
                    new_hosts = existing_hosts + f"\n127.0.0.1 {site_name}"
                    # Update the hosts file with the new content.
                    winreg.SetValueEx(key, "DataBasePath", 0,
                                      winreg.REG_SZ, new_hosts)
        except Exception as e:
            print("Unable to update hosts file:", e)

# Enable the WordPress site by starting the Docker containers.
def enable_site(site_name):
    subprocess.run(['docker-compose', 'start'])
    print(f"WordPress site '{site_name}' is enabled.")

# Disable the WordPress site by stopping the Docker containers.
def disable_site(site_name):
    subprocess.run(['docker-compose', 'stop'])
    print(f"WordPress site '{site_name}' is disabled.")

# Delete the WordPress site by stopping containers and removing the docker-compose.yml file.
def delete_site(site_name):
    subprocess.run(['docker-compose', 'down', '-v'])
    docker_compose_path = os.path.join(os.getcwd(), 'docker-compose.yml')
    # Remove the docker-compose.yml file after stopping the containers.
    os.remove(docker_compose_path)
    print(f"WordPress site '{site_name}' is deleted.")

# Main function to handle different subcommands and execute the corresponding actions.
def main():
    if len(sys.argv) < 3:
        print("Usage: python script.py <subcommand> <site_name>")
        sys.exit(1)

    subcommand = sys.argv[1].lower()
    site_name = sys.argv[2]

    # Check if Docker and docker-compose are installed or install them if necessary.
    check_docker()

    if subcommand == 'create':
        # Create a new WordPress site and add it to the hosts file.
        create_wordpress_site(site_name)
        add_hosts_entry(site_name)
        print(f"WordPress site '{site_name}' is ready at http://{site_name}/")
        open_browser = input("Do you want to open it in the browser? (y/n): ")
        if open_browser.lower() == 'y':
            webbrowser.open(f'http://{site_name}/')

    elif subcommand == 'enable':
        # Enable an existing WordPress site by starting the containers.
        enable_site(site_name)

    elif subcommand == 'disable':
        # Disable an existing WordPress site by stopping the containers.
        disable_site(site_name)

    elif subcommand == 'delete':
        # Delete an existing WordPress site, stopping containers, and removing files.
        delete_site(site_name)
    else:
        # If an invalid subcommand is provided, print the supported subcommands.
        print("Invalid subcommand. Supported subcommands: create, enable, disable, delete.")
        sys.exit(1)

# Execute the main function when the script is run.
if __name__ == "__main__":
    main()
```

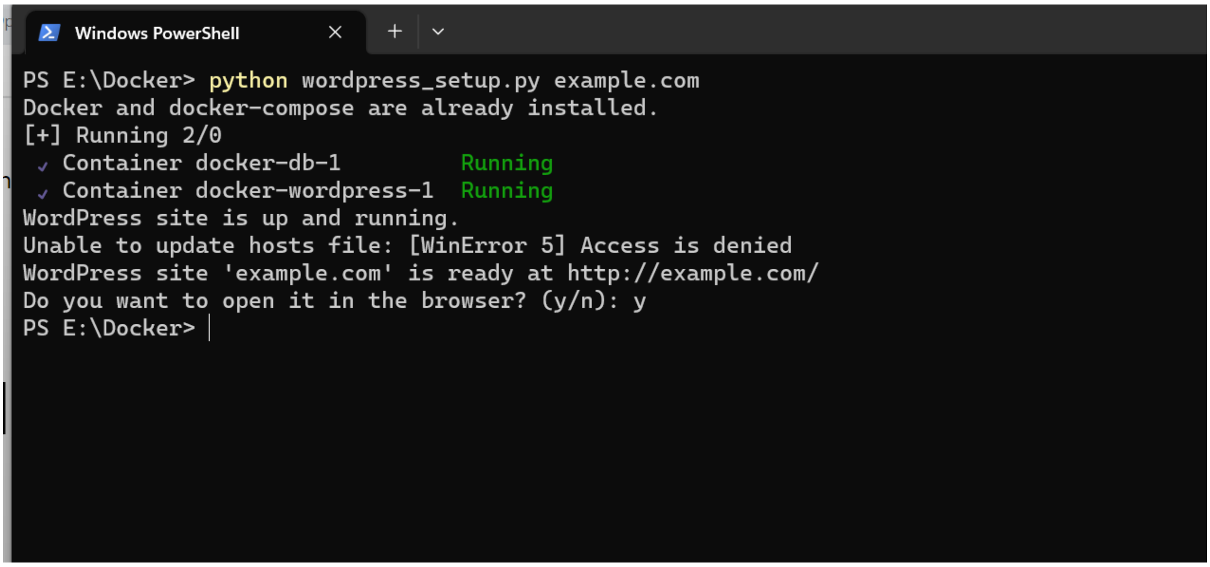
Output

1. Checking if docker is installed or not



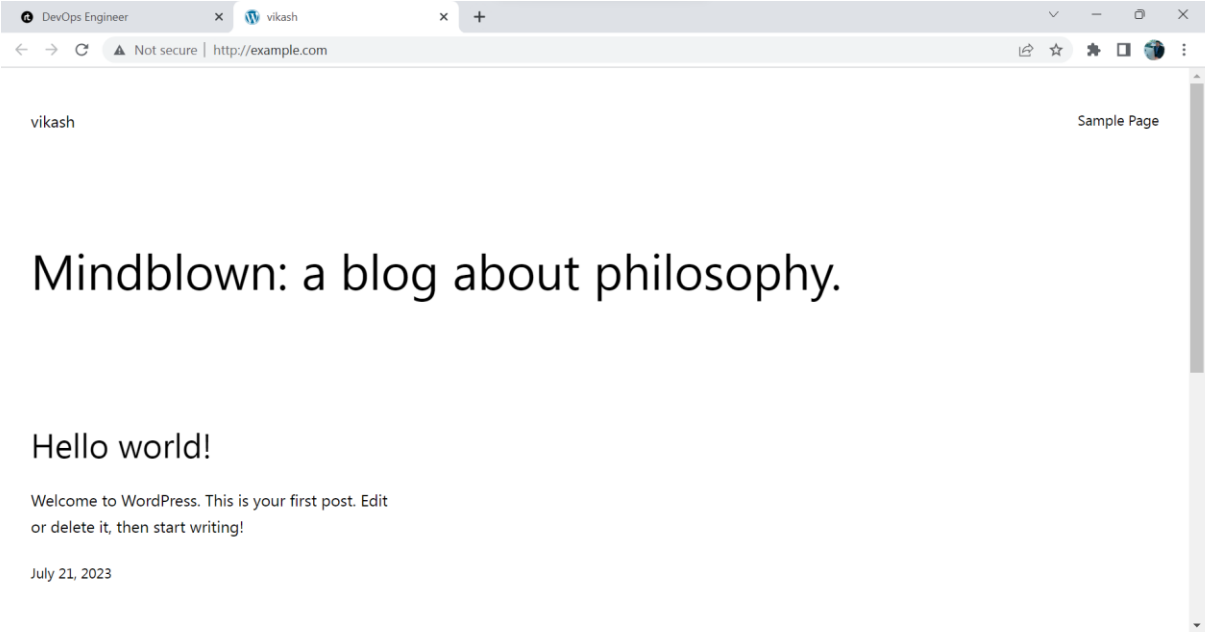
```
E:\Docker>python wordpress_setup.py example.com
Docker and docker-compose are already installed.
[+] Running 5/12
 - db 11 layers [⣀ ⣿⣿⣿⣿⣀⣿   ] 42.09MB/163.7MB Pulling                                                                 31.9s
   - 70e9ff4420fb Downloading [==============================>                    ]  31.04MB/50.48MB                 26.3s
   - 7ca4383b183f Verifying Checksum                                                                                26.3s
   ✓3e282e7651b1 Download complete                                                                                   2.0s
   ✓1ffa0e0ca707 Download complete                                                                                   7.7s
   ✓6eb790cf6382 Download complete                                                                                   4.0s
   ✓2b7ffc37d8e9 Download complete                                                                                   6.9s
   - 4393c12228b9 Downloading [===================>                               ]  9.968MB/25.53MB                 26.3s
   ✓389d2c130d52 Download complete                                                                                   9.0s
   - e5df3caef94c Downloading [>                                                  ]  1.081MB/87.66MB                 26.3s
   - 5c6aa409290d Waiting                                                                                           26.3s
   - faa350980ea9 Waiting                                                                                           26.3s
```

2. Prompting user to whether to open wordpress site or not



```
PS E:\Docker> python wordpress_setup.py example.com
Docker and docker-compose are already installed.
[+] Running 2/0
 ✓ Container docker-db-1          Running
 ✓ Container docker-wordpress-1   Running
WordPress site is up and running.
Unable to update hosts file: [WinError 5] Access is denied
WordPress site 'example.com' is ready at http://example.com/
Do you want to open it in the browser? (y/n): y
PS E:\Docker>
```

Website with URL : example.com



vikash

Sample Page

Mindblown: a blog about philosophy.

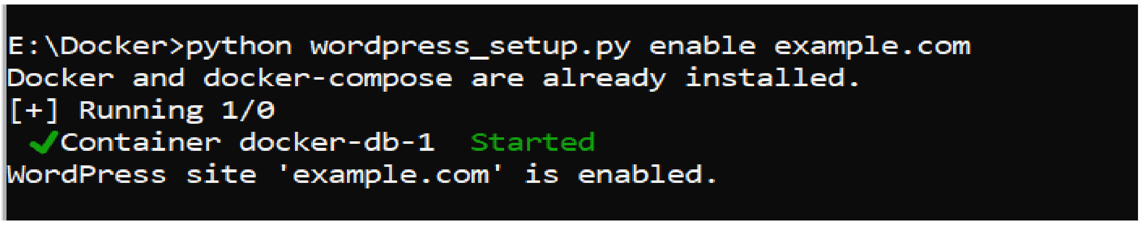
Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

July 21, 2023

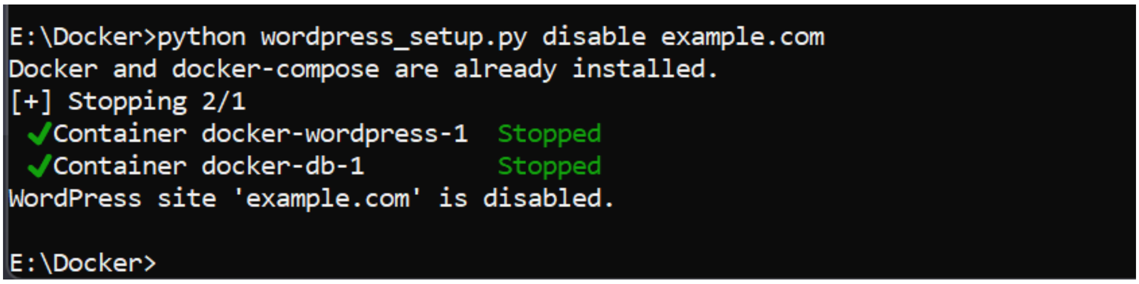
Add another subcommand to enable/disable the site (stopping/starting the containers)

1. Enable



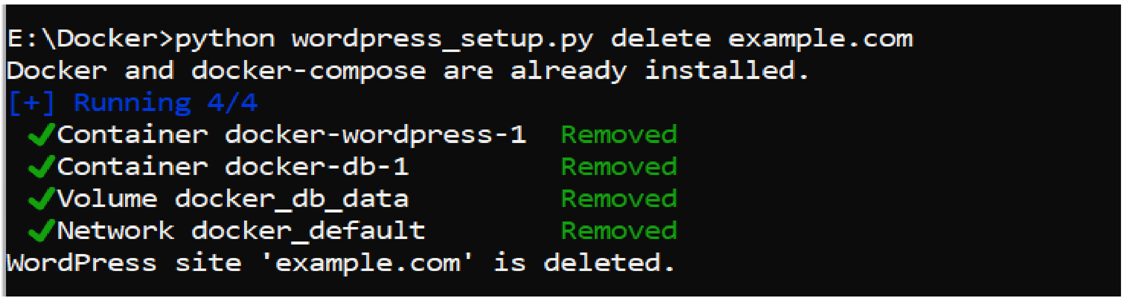
```
E:\Docker>python wordpress_setup.py enable example.com
Docker and docker-compose are already installed.
[+] Running 1/0
 ✓Container docker-db-1  Started
WordPress site 'example.com' is enabled.
```

2. Disable



```
E:\Docker>python wordpress_setup.py disable example.com
Docker and docker-compose are already installed.
[+] Stopping 2/1
 ✓Container docker-wordpress-1  Stopped
 ✓Container docker-db-1         Stopped
WordPress site 'example.com' is disabled.

E:\Docker>
```

3. Delete



```
E:\Docker>python wordpress_setup.py delete example.com
Docker and docker-compose are already installed.
[+] Running 4/4
 ✔Container docker-wordpress-1  Removed
 ✔Container docker-db-1         Removed
 ✔Volume docker_db_data         Removed
 ✔Network docker_default        Removed
WordPress site 'example.com' is deleted.
```