

# Development Operations

## Mini Project Assignment # 1

REG#: \_\_\_\_\_

NAME: \_\_\_\_\_

COURSE CODE: CS423

INSTRUCTOR: MUHAMMAD SAJID ALI

TOTAL MARKS: 314

---

### Comprehensive DevOps, Linux System Administration, and AWS Cloud

#### Objective

This assignment is designed to provide you with practical, hands-on experience in essential aspects of DevOps, Linux system administration, and AWS Cloud services. Throughout this assignment, you will perform a series of tasks aimed at building foundational skills and knowledge in these areas. Proper documentation is crucial for demonstrating your understanding and completing the tasks successfully.

#### General Instructions:

Follow the instructions provided for each task and subtask carefully. Ensure that you complete each task and subtasks in the order specified. By following these guidelines, you will organize your work effectively and provide clear documentation for each task. This approach will help you complete the assignment successfully and streamline the reporting process.

#### Capturing Screenshots

- Take screenshots of your terminal or any graphical interface as required by the tasks.
- Name the screenshot files descriptively, such as *taskX.Y.png*, where X is the task number, and Y is the subtask number.

#### Saving Commands and Outputs

- Create a text file named *commands.txt* for each task inside the appropriate directory.
- In this file, list all the commands you executed in the order they were executed.
- Additionally, document any command outputs (redirections) as asked in the task or needed in a separate file descriptively, such as *taskX.Y\_output.txt*, where X is the task number, and Y is the subtask number.

#### Organizing Files

- Create a directory structure in your home folder for this assignment. Each task should have its own directory (e.g., *task1*, *task2*, etc.).
- Within each task directory, create subdirectories for each subtask where needed (e.g., *part1*, *part2*, etc.).
- Place the corresponding screenshot files, *commands.txt*, and any other relevant files in the appropriate task or subtask directories.

#### Verification Screenshots

- Some subtasks may require you to verify your work by running specific commands.
- Capture screenshots of these verification steps and save them with a name that reflects the purpose (e.g., *taskX.Y\_verification.png*).

#### Using Consistent Filenames

- Maintain a consistent naming convention for your files to avoid confusion.
- Ensure that filenames reflect the task, subtask, and purpose.

**Task 1: Setting Up a Linux Environment on AWS.****(CLO1, PLO1) (50 marks)**

1. Create an IAM user named 'cs423-devops' through AWS IAM service. Provide console access and assign the Administrator policy to this user. Generate the user's credentials, as you will include them in your assignment report for the instructor to use for verification, if necessary. Take appropriate screen shots and save with appropriate name as guided in general instructions section. **(10 marks)**
2. Create VPC with following settings: **(20 marks)**
  - a. VPC named 'devops-assignment-1' with CIDR block of <address of your choice>/16.
  - b. Create two pairs of public and private subnets, one in each of two availability zones, with names like 'cs423-devops-public-1' and 'cs423-devops-private-1'. Each subnet should support up to 255 virtual machines.
  - c. Create a private route table with a name that starts with the VPC name and is followed by arbitrary text. Ensure that the private subnets are configured to have no access to the internet.
  - d. Create an internet gateway to give access to the resources deployed in your created VPC.
3. Launch two t2.micro EC2 instances within the previously established VPC. Use the latest Ubuntu Amazon Machine Image (AMI). Assign each instance to a specific subnet based on its intended purpose: one for hosting a website and the other for deploying a database instance or a simple machine learning model that communicates with the web application. **(20 marks)**
  - a. Give appropriate name to these instances such as 'Assignment1-EC2-1' etc.
  - b. Make sure when you hit launch instance it should update the apt sources for you to start configuring this server.

**Task 2: User and Group Management.****(CLO2, PLO5) (72 marks)**

In this task, you will practice user and group management on an Ubuntu system. You will create a user named cs423\_devops\_user, modify its properties, add it to a group, and perform other related tasks. Follow the instructions for each subtask, take screenshots, and save command outputs as needed in files labeled with corresponding subtask numbers under the main task directory in your home folder.

1. Using the command line, create five user accounts of the given pattern that you'll use for future tasks. These accounts will be used for testing permissions and ownership. Each user should have a different name but set them all to have the same password to keep things straightforward. **(10 marks)**
  - a. Username pattern: cs423\_devops\_user\_1, cs423\_devops\_user\_2 ...
2. Use the command line to make five groups for the given faculties: *fcse*, *fee*, *fes*, *fme*, and *fcve*. Put one of the users we created earlier into each group in an order. For example, assign the first user to the "fcse" group, the second user to the "fee" group, and so on. **(12 marks)**
3. Delegate the ability to reboot the system to the second user by using sudo. **(05 marks)**
4. Grant administrative privileges to the first user. **(05 marks)**
5. Generate key pairs for each user, naming them after their respective usernames. Users should be able to connect to a server using these key pairs and optionally password as well. **(15 marks)**
6. Create and configure an SSH connection setup for each user by managing a config file. This allows you to connect to the remote server by simply typing "ssh username," with "username" being the server's name. **(10 marks)**
7. Attach my key, which is included with this assignment, to your EC2 instance so that I can access your remote server successfully. **(05 marks)**
8. Ensure that all SSH connections are established on both port 22 and port 25. **(10 marks)**

**Task 3: Managing Files and Directories.****(CLO2, PLO5) (82 marks)**

In this task, you will continue to work on an Ubuntu system where you have already created user accounts and groups. You will use these users and groups to manage files and directories, set appropriate permissions, and test ownership and access controls. Follow the instructions for each subtask, take screenshots, and save command outputs as needed in files labeled with corresponding subtask numbers under the main task directory in your home folder.

1. Create directories named after each faculty (fcse, fee, fes, fme, fcve) at the root of the filesystem. **(10 marks)**
2. Set up standard *Linux permissions* and *ownership* so that each group can only access its respective department's directory. For instance, the fcse group should have access to the /fcse directory but not to directories of other departments. **(15 marks)**
  - a. Once you've finished and tested the previous step, use *ACL permissions* to allow the fcse faculty to read the contents of all other faculty directories.
  - b. Use ACL permissions to grant read access to the /fcve directory for a *third* user account.
3. Using different user accounts, create files within recently created directories. Observe how permissions impact file creation. Must write your observation as an explanation in a file and place it in the same folder as you have been doing for other tasks screenshots and executed commands. **(15 marks)**
4. Use the find command to search for test files in department directories three times: as the root user, as a group member, and as a non-group member. Write your observation with screenshots. **(12 marks)**
5. Create a directory named 'task3-4' under your home directory. In this newly created directory, use attached Bash script that will create various files with different timestamps and sizes. Use this environment to complete the following tasks: **(30 marks)**
  - a. Use *tar* and *gzip* to create compressed bundles of all the files created by the bash script.
  - b. Delete the created files and restore them from the compressed bundle.
  - c. Using *find* command do the following tasks:
    - i. Find files modified within the last 3 months.
    - ii. Find files larger than 100KB.
    - iii. Find files whose names have the digit '3' in it.
    - iv. Find and delete files larger than 100KB and older than 2 months.
    - v. Find and delete files created within the last 10 days.
    - vi. Find and delete files smaller than 1KB and older than 4 months.

**Task 4: Updating and Installing Packages.****(CLO2, PL05) (CLO4, PL05) (60 marks)**

1. Update the package inventory to ensure you have the latest package lists and redirect this inventory list into a text file name 'task4-1.txt' in your-home-dir/task4 directory. **(05 marks)**
2. Show the details of the nano and save the output text file name 'task4-2.txt' in your-home-dir/task4 directory. **(05 marks)**
3. Upgrade the nano editor. **(50 marks)**
4. Our next target is to install WordPress on your AWS instance, for this follow the following instructions.
  - a. Install Apache web server.
  - b. Install PHP 7.4 or greater.
  - c. Install MySQL 5.7 or MariaDB 10.4 or higher.
  - d. Display the document root path of the Apache server.
  - e. Download the adminer.php file from [here](#) and copy it to the root path of the Apache server.
  - f. Go to your browser and hit <http://your-server-ip/adminer-file-name.php> to setup database for WordPress site.
  - g. Download and extract the WordPress from [this](#) link inside your EC2 instance.
  - h. Copy all the extracted files to the root path of Apache server.
  - i. Go to your browser and hit <http://your-server-ip/>. This will start the installation wizard of the WordPress site.
  - j. You should be able to visit your first deployed wordpress site by visiting the url: <http://your-server-ip/>

**Task 5: Monitoring Resource Performance and Status.****(CLO1, PL01) (CLO2, PL05) (50 marks)**

Use the following commands to document hardware information for your EC2 instance: df, du, cat /proc/cpuinfo, cat /proc/meminfo, devices, partitions, ifconfig, ip addr, cat /etc/fstab, uname -a. Direct the output from these commands into text files (make as many text files as you wish to organize this documentation reasonably). Follow the instructions

for each task, take screenshots, and save command outputs as needed in files labeled with corresponding subtask numbers under the main task directory in your home folder. Use both > and >> to organize the data.

1. List selection of or all running processes and their PIDs. **(05 marks)**
2. Sleep for 10 s and run this sleep process in background. **(05 marks)**
3. List the background processes. **(05 marks)**
4. List the active processes and their resource usage. **(05 marks)**
5. List mounted file systems and their usage in human readable form. This should also include the total available space. **(05 marks)**
6. List mounted file systems and their usage to file named '<your\_reg\_number>\_mfsu.txt'. This should include the following details only. [Hint, try to make use of awk utility] **(20 marks)**
  - a. File system name
  - b. Total size
  - c. Total usage in %
7. Display the free memory in human readable form. **(05 marks)**

### Submission

1. Use tar and gzip to create compressed bundle of all inside your home directory under name assignment1\_<your\_reg\_number>.zip
2. Upload the tarball file on teams.