St. Francis Institute of Technology, Mumbai-400 103

**Department Of Information Technology**

A.Y. 2024-2025

Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

**Experiment – 1: To understand DevOps: Principles, practices and DevOps Engineer role & responsibilities and learn basic Linux commands**

1. **Aim:** To prepare case study on DevOps and execute Linux commands
2. **Objectives:** After study of this experiment, the students will be able to
   * Understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits and deployment options to meet business requirements
   * Demonstrate execution of Linux commands
3. **Outcomes:** After study of this experiment, the students will be able to
   * Understand the importance of DevOps thoroughly
   * Demonstrate the Linux commands
4. **Prerequisite:** Knowledge of software engineering and project management basic principles
5. **Requirements:** Personal Computer, Windows operating system, Ubuntu Operating system or Oracle virtual box or any online terminal, Internet Connection, Microsoft Word.
6. **Pre-Experiment Exercise:**

**Brief Theory:** Refer shared material

1. **Laboratory Exercise**
   * + 1. **Procedure:**

**a.** Prepare case study based on following:

* Introduction
* Need
* Usage
* DevOps Lifecycle
* Advantages and disadvantages

**b**. **Enlist following Linux commands with their explanation and execution screenshots**

1. sudo
2. Apt -get
3. Ls
4. Cd
5. Ped
6. Cp
7. Mv
8. Rm
9. Mkdir
10. History
11. Df
12. Du
13. Free
14. Uname -a
15. Top
16. Man
17. Info
18. Passwd
19. Whatis
20. Date
21. W
22. Exit
23. Shutdown
24. Head
25. Tail
26. Echo
27. Grep
28. Zip
29. Unzip
30. <command name> -h or <command name> –help
31. **Post-Experiments Exercise**
32. **Extended Theory:**

Nil

1. **Questions:**

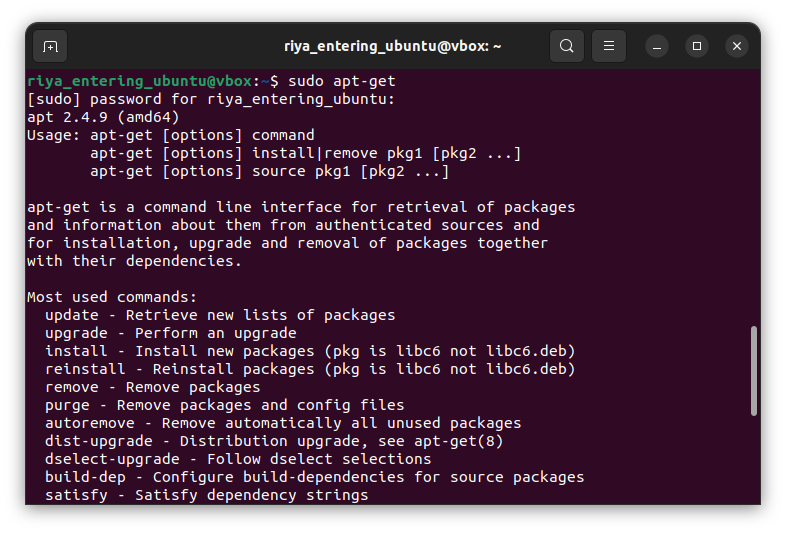
* What is Linux and basic components of Linux?
* What are the shells used in Linux?
* What are the top 10 devops tools that are used in the industry today?
* How will you approach when a project needs to implement devops?

1. **Conclusion:**

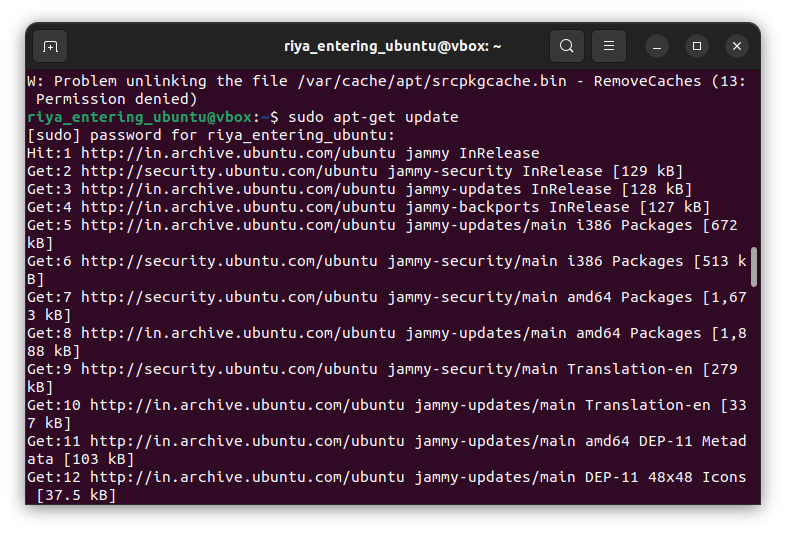
* Write what was performed in the experiment.
* Write the significance of the topic studied in the experiment.

1. **References:**

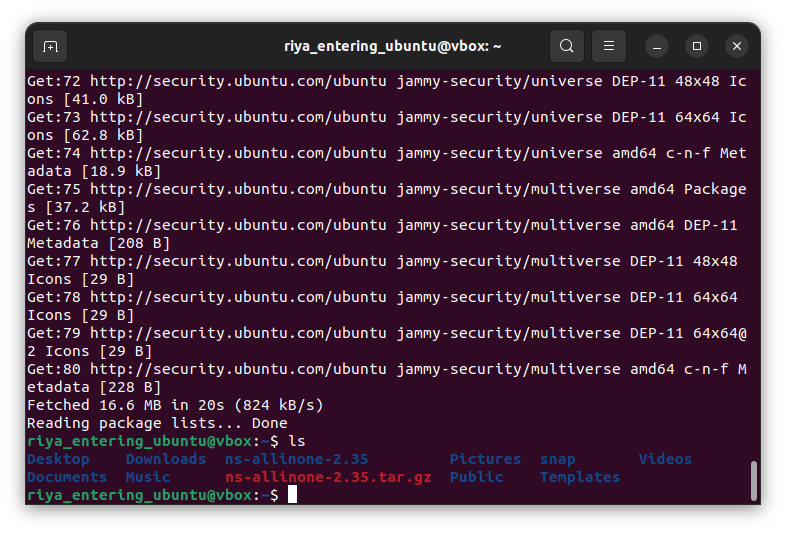
* <https://www.edureka.co/blog/interview-questions/linux-interview-questions-for-beginners/>
* https://www.softwaretestinghelp.com/devops-interview-question11
* <https://techlog360.com/basic-ubuntu-commands-terminal-shortcuts-linux-beginner/>
* https://tutorials.ubuntu.com/tutorial/command-line-for-beginners#0
* https://techlog360.com/basic-ubuntu-commands-terminal-shortcuts-linux-beginner/
* https://www.edureka.co/blog/top-10-devops-tools/
* https://www.guru99.com/devops-tutorial.html

**sudo**:sudo (SuperUser DO) Linux command allows you to run programs or other commands with administrative privileges, just like “Run as administrator”in Windows.

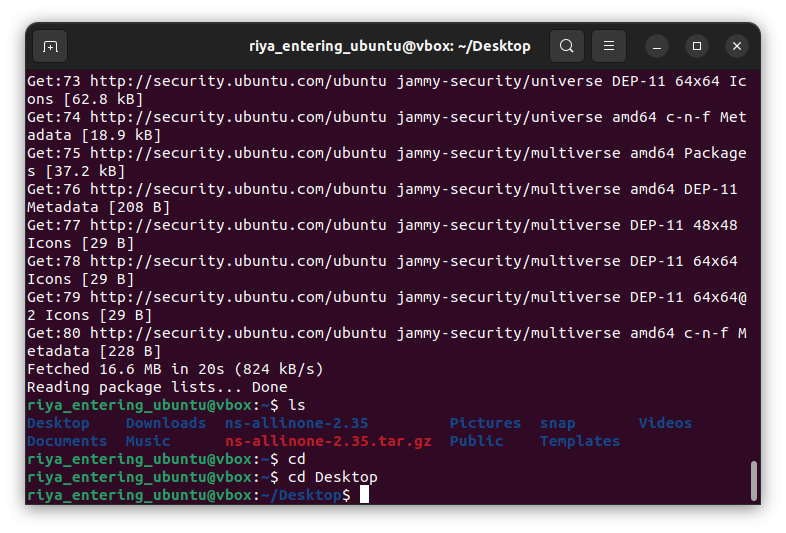
**apt-get**:apt-get is the one of the most important Ubuntu commands.It is used to install, update, upgrade and remove any package. apt-get basically works on a database of available packages.we can do update,install,remove,etc



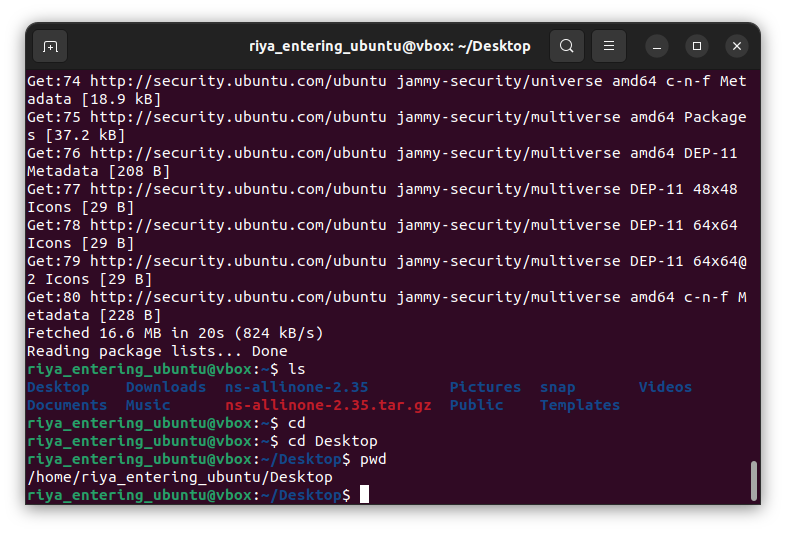
**ls**:ls (list) command lists all files and folders in your current working directory.



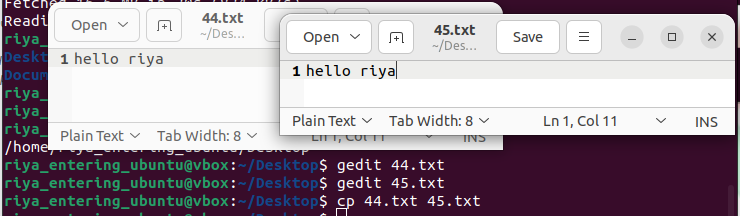
**cd**:cd (change director”) Linux command also known as chdir used to changethe current working directory.



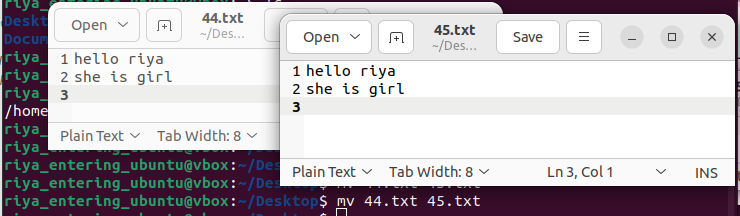
**pwd**:pwd (print working directory) Ubuntu command displays the full pathname of the current working directory.



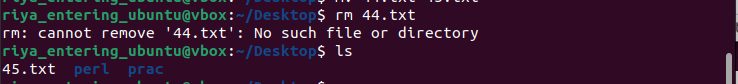
**cp**:cp (copy) Linux command allows you to copy a file.



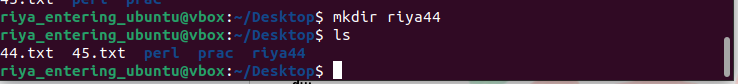
**mv**:mv (move) command allows you to move files.



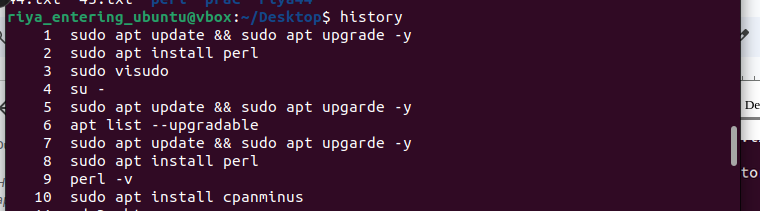
**rm**:rm (remove) command removes the specified file.



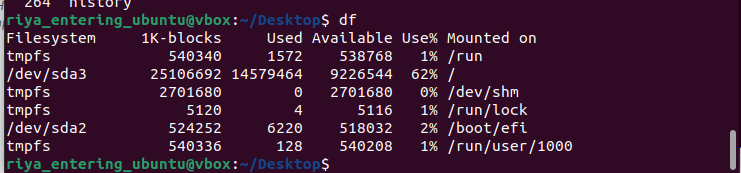
**mkdir**:mkdir (make directory) command allows you to create a new directory.



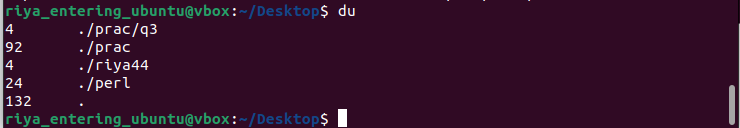
**history**:history command displays all of your previous commands up to the history limit.



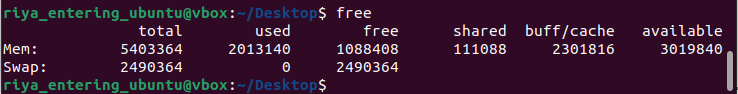
**df**:df (display filesystem) command displays information about the disk space usage of all mounted filesystems.



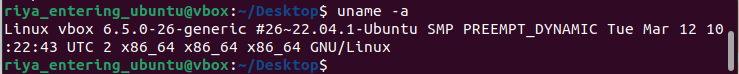
**du**:du (directory usage) command displays the size of a directory and all of its subdirectories.



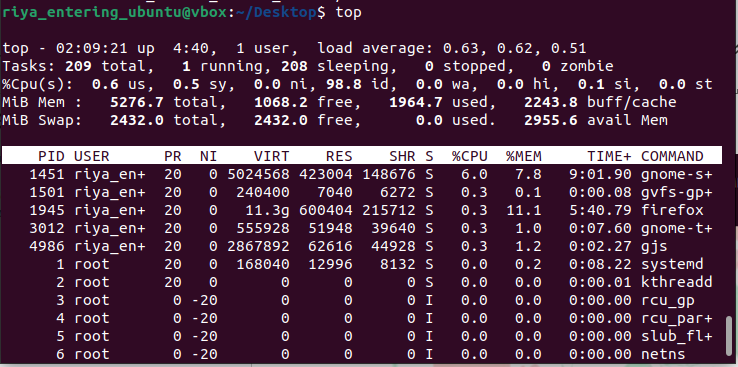
**free**:– Displays the amount of free space available on the system.



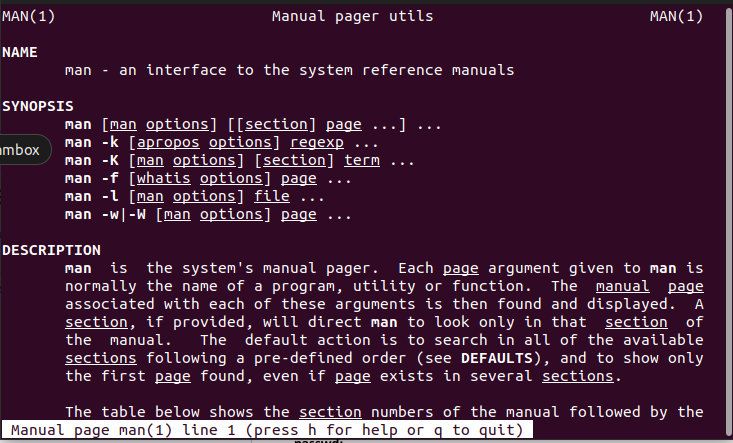
**uname -a**:Provides a wide range of basic information about the system.



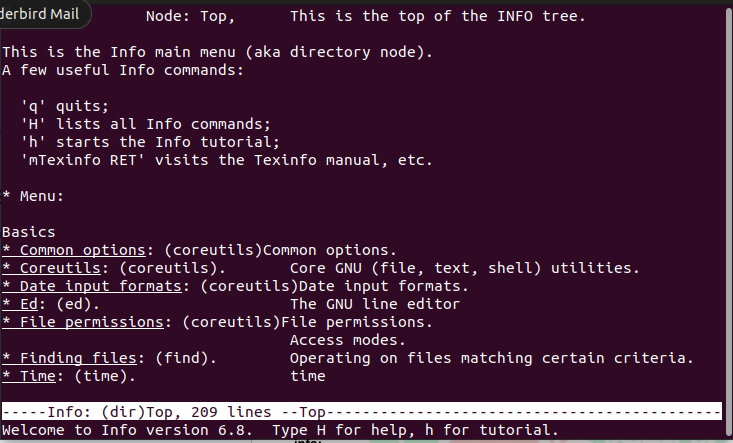
**top**:– Displays the processes using the most system resources at any given time. “q” can be used to exit.



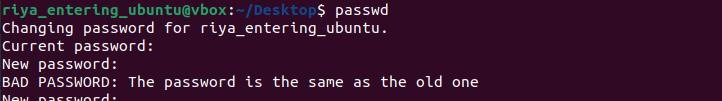
**man**:man command displays a “manual page”. Manual pages are usually very detailed.



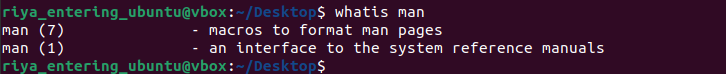
**info**:It is Similar to man, but often provides more detailed or precise information.



**passwd**:basic command is used to change user password using Terminal.



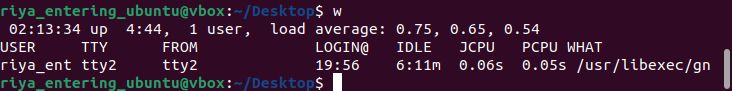
**whatis**:whatis command shows a brief description of what is the functionality of specific built-in Linux command.



**date**:whatis command shows a brief description of what is the functionality of specific built-in Linux command.

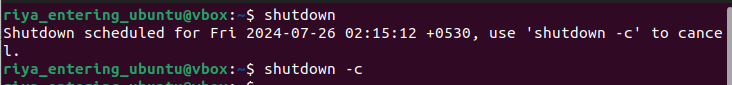


**w**:The command “w” displays the detailed information about the users who are logged in the system currently.

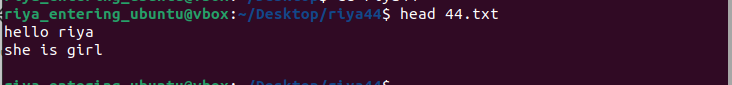


**exit**:The command “exit” as the name says it is used to exit from the system and log out from the current user.

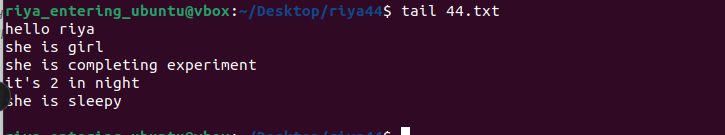
**shutdow**n:The command “shutdown” is used to shut down the system.



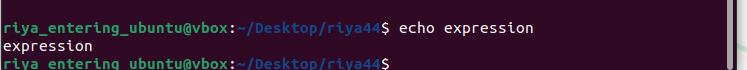
**head**:prints the top N rows of data of the given input or file. By default, it prints the first 10 lines of the specified files.



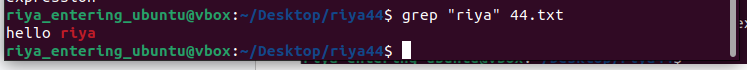
**tai**l:prints the last N rows of data of the given input or file. By default, it prints the last 10 lines of the specified files.



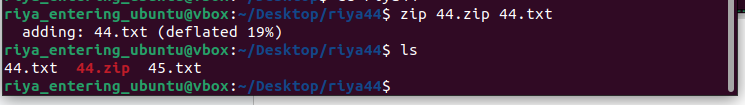
**echo**:“echo” used to display any expression that is passed as an argument.



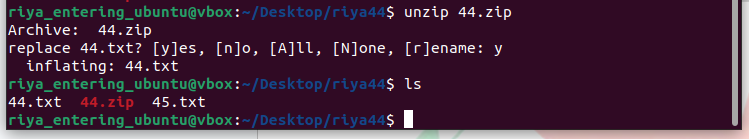
**grep**:is used to search for a text in the specified file/folder.



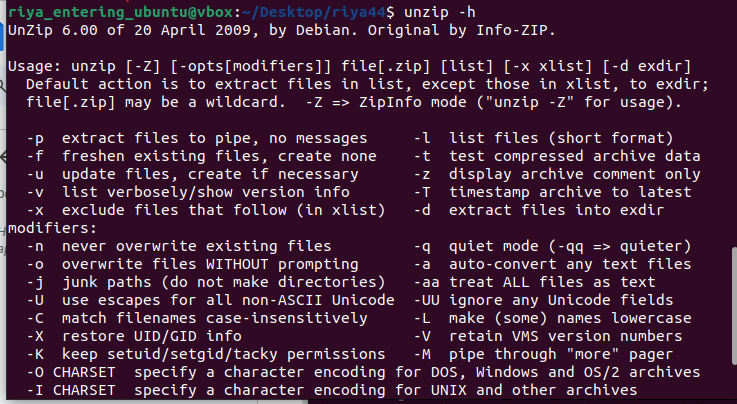
**zip**:used to compress one or more files and store them in a new file with .zip extension.



**unzip**:used to decompress a .zip file and extract all the files within to current directory.



unzip -h/help:gives details about the command mentioned.



### Case Study: Implementing DevOps in a Software Development Organization

#### 1. Introduction

In the contemporary software development landscape, the need for speed, quality, and efficiency has never been greater. Traditional development methodologies often struggle to meet these demands, leading many organizations to adopt DevOps practices. DevOps integrates development (Dev) and operations (Ops) teams to streamline the software development lifecycle (SDLC), enhance collaboration, and improve the overall delivery process.

This case study explores the implementation of DevOps in a medium-sized software development organization, focusing on its need for DevOps, its usage, the lifecycle it follows, and the advantages and disadvantages it encountered.

#### 2. Need

The organization, a mid-sized company specializing in enterprise software solutions, faced several challenges:

* **Slow Deployment Cycles:** Releases were infrequent and took several weeks to months, delaying the delivery of new features and bug fixes.
* **Poor Collaboration:** Development and operations teams operated in silos, leading to communication breakdowns and inefficiencies.
* **High Failure Rates:** Manual deployments and lack of automated testing resulted in frequent errors and rollbacks.
* **Customer Demands:** Increasing customer expectations for rapid updates and continuous improvements highlighted the need for a more agile approach.

To address these issues, the organization recognized the need for a DevOps transformation to streamline workflows, enhance collaboration, and accelerate deployment cycles.

#### 3. Usage

The organization implemented DevOps using the following approach:

* **Adoption of DevOps Tools:** They integrated tools such as Jenkins for Continuous Integration (CI), Docker for containerization, Kubernetes for orchestration, and Nagios for monitoring.
* **Automation:** Automated build, test, and deployment processes were introduced to replace manual tasks. This included automated unit and integration tests to catch issues early.
* **CI/CD Pipeline:** A robust Continuous Integration/Continuous Deployment (CI/CD) pipeline was established, allowing for frequent and reliable releases.
* **Collaboration Practices:** Regular cross-functional team meetings were introduced, fostering improved communication between development, operations, and quality assurance (QA) teams.
* **Monitoring and Feedback:** Implemented monitoring and logging tools to provide real-time feedback and performance metrics, facilitating proactive issue resolution.

#### 4. DevOps Lifecycle

The DevOps lifecycle in the organization follows these stages:

* **Planning:** Define requirements, set goals, and plan the project with input from both development and operations teams.
* **Development:** Develop code with continuous integration processes, ensuring code changes are frequently integrated and tested.
* **Testing:** Automated testing ensures that code changes do not introduce new bugs or issues. Testing environments are managed through containerization to ensure consistency.
* **Deployment:** Automated deployment processes are executed using CI/CD pipelines, enabling frequent and reliable releases.
* **Operations:** Post-deployment, operations teams monitor applications and infrastructure, addressing issues as they arise and ensuring system stability.
* **Monitoring:** Continuous monitoring and feedback loops are used to gather performance metrics and user feedback, informing future development and operational improvements.

#### 5. Advantages and Disadvantages

**Advantages:**

* **Faster Time-to-Market:** Automated processes and CI/CD pipelines significantly reduce deployment times, allowing for quicker release of features and bug fixes.
* **Improved Quality:** Automated testing and continuous integration help in identifying and resolving issues early in the development process, leading to higher quality releases.
* **Enhanced Collaboration:** DevOps practices foster better communication and collaboration between development and operations teams, reducing silos and improving efficiency.
* **Increased Stability:** Automated monitoring and proactive issue resolution contribute to higher system reliability and stability.
* **Scalability:** Containerization and orchestration with tools like Kubernetes facilitate easier scaling of applications and infrastructure.

**Disadvantages:**

* **Initial Costs:** Implementing DevOps requires an upfront investment in tools, training, and infrastructure, which can be a significant expense.
* **Complexity:** The transition to DevOps can be complex and challenging, requiring changes to existing workflows, processes, and team structures.
* **Cultural Shift:** Adopting DevOps involves a cultural shift that may meet resistance from employees accustomed to traditional practices and workflows.
* **Ongoing Maintenance:** Continuous monitoring and maintenance of DevOps tools and pipelines require ongoing effort and resources.