St. Francis Institute of Technology, Mumbai-400 103

Department Of Information Technology

A.Y. 2023-2024

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

1. **Prerequisite:** Knowledge of software engineering and project management basic principles
2. **Requirements:** Personal Computer, Windows operating system, Ubuntu Operating system or Oracle virtual box or any online terminal, Internet Connection, Microsoft Word.
3. **Pre-Experiment Exercise:**

**Brief Theory:** Refer shared material

1. **Laboratory Exercise**

**A. Procedure:**

1. Prepare case study based on following:
   * Introduction
   * Need
   * Usage
   * DevOps Lifecycle
   * Advantages and disadvantages
2. **Enlist following Linux commands with their explanation and execution screenshots**
3. sudo
4. Apt -get
5. Ls
6. Cd
7. Pwd
8. Cp
9. Mv
10. Rm
11. Mkdir
12. History
13. Df
14. Du
15. Free
16. Uname -a
17. Top
18. Man
19. Info
20. Passwd
21. Whatis
22. Date
23. W
24. Exit
25. Shutdown
26. Head
27. Tail
28. Echo
29. Grep
30. Zip
31. Unzip
32. <command name> -h or <command name> –help
33. **Post-Experiments Exercise**
    1. **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?
  2. **Conclusion:**
     + Write what was performed in the experiment.
     + Write the significance of the topic studied in the experiment.
  3. **References:**
* [https://www.edureka.co/blog/interview-questions/linux-interview-questions-for-be](https://www.edureka.co/blog/interview-questions/linux-interview-questions-for-beginners/) [ginners/](https://www.edureka.co/blog/interview-questions/linux-interview-questions-for-beginners/)
* https://[www.softwaretestinghelp.com/devops-interview-question11](http://www.softwaretestinghelp.com/devops-interview-question11)
* [https://techlog360.com/basic-ubuntu-commands-terminal-shortcuts-linux-begin](https://techlog360.com/basic-ubuntu-commands-terminal-shortcuts-linux-beginner/) [ner/](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-begin ner/
* https://[www.edureka.co/blog/top-10-devops-tools/](http://www.edureka.co/blog/top-10-devops-tools/)
* https://[www.guru99.com/devops-tutorial.html](http://www.guru99.com/devops-tutorial.html)



**Case Study: Implementing DevOps Lifecycle for a Web Application**

# Case Study

DevOps is a culture and a set of practices that combines software development (Dev) and IT operations (Ops) to shorten the systems development life cycle while delivering high quality software. It aims at establishing a culture and environment where building, testing, and releasing software can happen rapidly, frequently, and more reliably.

DevOps teams are typically cross-functional, with members from both development and operations working together to deliver software. This helps to break down silos and ensure that everyone is working towards the same goals.

## DevOps is based on the following principles:

◉ Culture: DevOps is more than just a set of tools and processes. It is a cultural shift that requires organizations to break down silos and create a more collaborative environment between development and operations teams.

◉ Automation: DevOps relies heavily on automation to streamline the software delivery process. This can be done using tools like continuous integration and continuous delivery.

◉ Monitoring: DevOps teams use monitoring tools to track the performance of their applications and infrastructure. This helps them to identify and fix problems quickly.

◉ Communication: Effective communication is essential for DevOps teams. They need to be able to communicate effectively with each other, as well as with other stakeholders in the organization.

## Need:

◉ To keep up with the pace of change: The world of software development is constantly changing, and organizations need to be able to keep up in order to stay competitive. DevOps can help organizations to do this by automating the software delivery process and by making it easier to release new features and updates.

◉ To improve the quality of software: DevOps can help organizations to improve the quality of their software by automating the testing process and by making it easier to roll back changes that are not working as expected. This can lead to fewer bugs, less downtime, and happier customers.

◉ To reduce costs: DevOps can help organizations to reduce the costs of software development and infrastructure management by automating tasks and by eliminating the need for manual intervention. This can free up resources that can be used to focus on other areas of the business.

◉ To improve collaboration: DevOps can help to improve collaboration between development and operations teams. This can lead to a better understanding of each other's needs and a more efficient software delivery process.

◉ To create a more agile organization: DevOps can help organizations to become more agile and responsive to change. This can give organizations a competitive advantage in today's rapidly changing business environment.

## Usage:

◉ Technology: DevOps is widely used in the technology industry, where organizations need to be able to quickly release new features and updates to their software.

◉ Finance: DevOps is also used in the financial industry, where organizations need to be able to maintain the security and stability of their systems.

◉

Retail: DevOps is also used in the retail industry, where organizations need to be able to handle large volumes of traffic and transactions.

◉ Healthcare: DevOps is also used in the healthcare industry, where organizations need to be able to quickly deploy new medical devices and software.

◉ Manufacturing: DevOps is also used in the manufacturing industry, where organizations need to be able to quickly deploy new products and services.

|  |  |
| --- | --- |
|  |  |

## DevOPs Lifecycle:

The DevOps lifecycle can be divided into the following phases:

◉ Planning: In the planning phase, the team defines the scope of the project, identifies the requirements, and creates a plan for development and deployment.

◉ Development: In the development phase, the team writes the code, tests the code, and fixes any bugs.

◉ Testing: In the testing phase, the team tests the code for functionality, performance, and security.

◉ Deployment: In the deployment phase, the team deploys the code to production.

◉ Monitoring: In the monitoring phase, the team monitors the performance of the code in production and identifies any problems.

◉ Reporting: In the reporting phase, the team generates reports on the performance of the code in production.

## Advantages and disadvantages:

**Advantages:**

◉ Increased speed of delivery: DevOps can help organizations to deliver new features and updates to their software more quickly. This is because DevOps automates the process of development, testing, and deployment.

◉

Improved quality: DevOps can help organizations to improve the quality of their software by incorporating testing into the development process. This can help to identify and fix bugs early on, which can lead to improved quality.

◉ Reduced costs: DevOps can help organizations to reduce the costs of software development and deployment by automating tasks and eliminating the need for manual intervention.

◉ Improved collaboration: DevOps can help to improve collaboration and communication between development and operations teams. This can lead to a more efficient software delivery process.

◉ Increased agility: DevOps can help organizations to be more agile and responsive to change. This is because DevOps teams are able to quickly deploy new features and updates to their software.

## Disadvantages:

◉ Cultural change: DevOps requires a cultural change in the organization, which can be difficult to achieve. This is because DevOps requires development and operations teams to work together closely, which can be challenging if they have traditionally worked in silos.

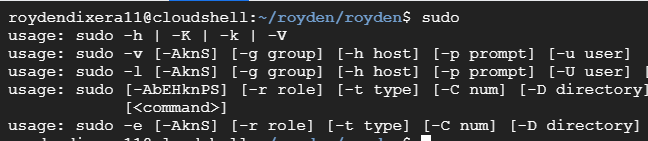
◉ Technical challenges: DevOps can be technically challenging to implement, especially in organizations with complex systems. This is because DevOps requires a number of different tools and technologies, and it can be difficult to integrate these tools and technologies together.

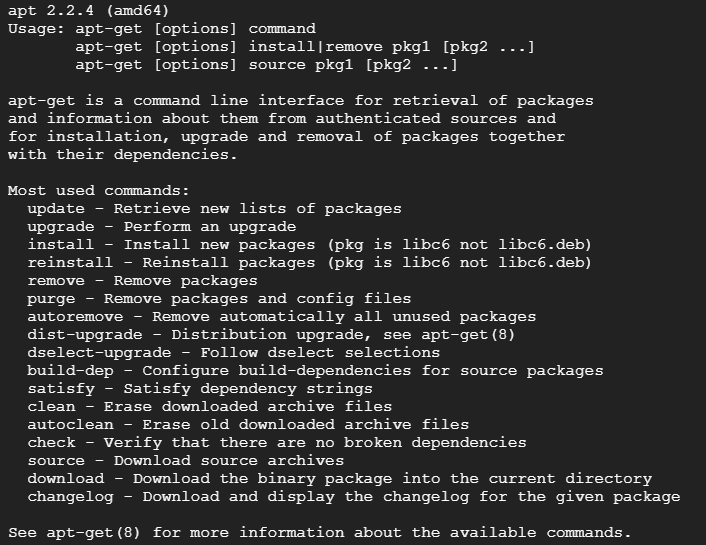
◉ Cost: DevOps can be expensive to implement, especially in organizations with large development and operations teams. This is because DevOps requires a number of different tools and technologies, and it can be expensive to purchase and maintain these tools and technologies.

◉ Security risks: DevOps can introduce new security risks, as it involves automating the process of development, testing, and deployment. This is because automation can make it easier for attackers to exploit vulnerabilities in software.

◉ Risk of failure: DevOps can be risky, as it involves implementing a number of changes to the software development process. This is because changes can introduce new problems, and it can be difficult to predict the impact of these changes.

1. **sudo**



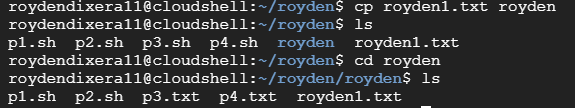
1. **Apt -get**
2. 
3. **Ls**
4. 
5. **Cd**



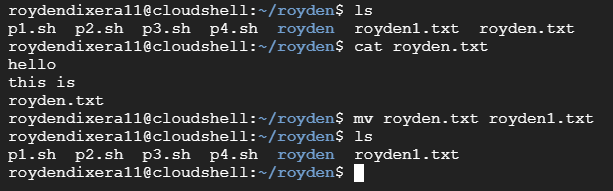
1. **Pwd**



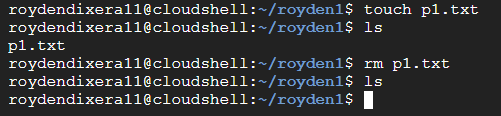
1. **Cp**



1. **Mv**



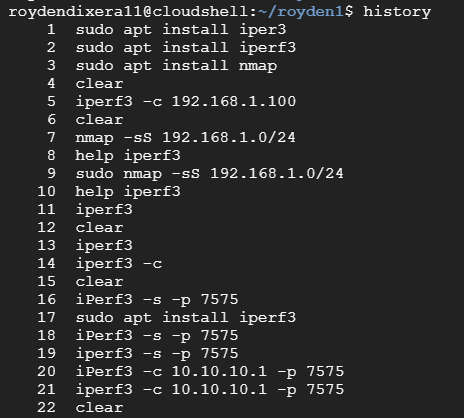
1. **Rm**



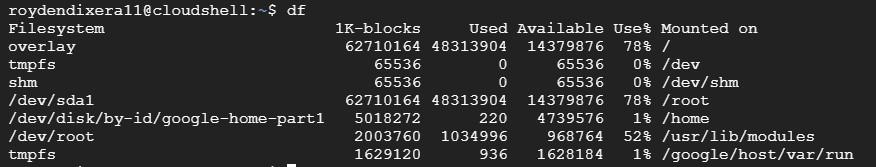
1. **Mkdir**



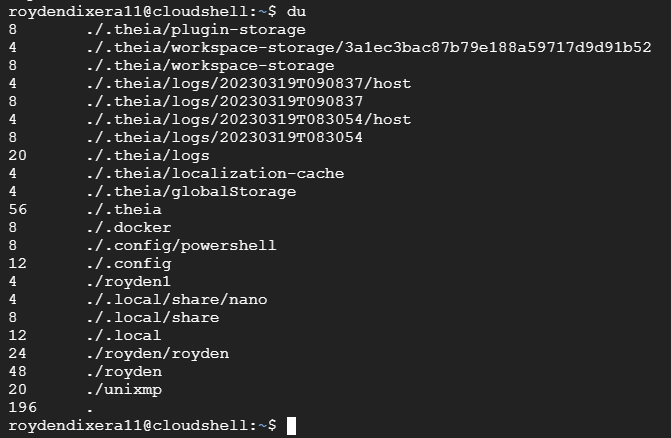
1. **History**



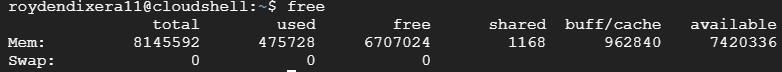
1. **Df**



1. **Du**



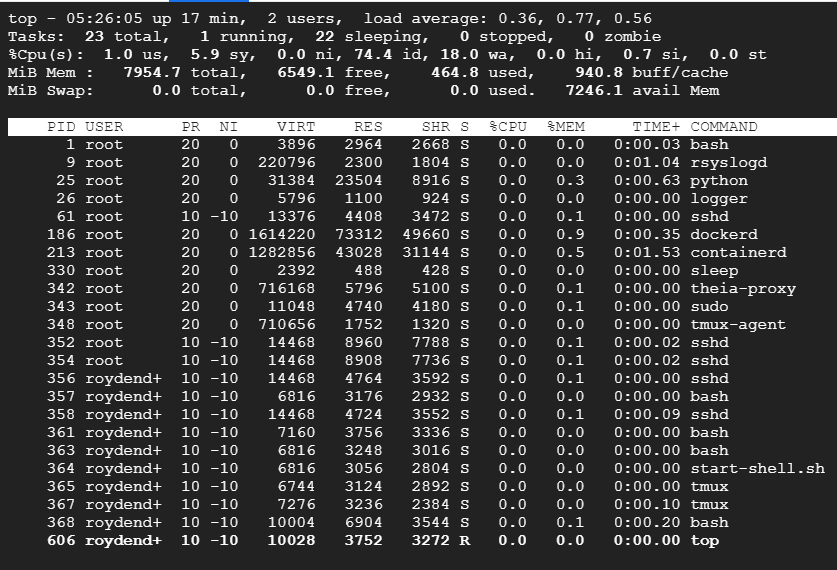
1. **Free**



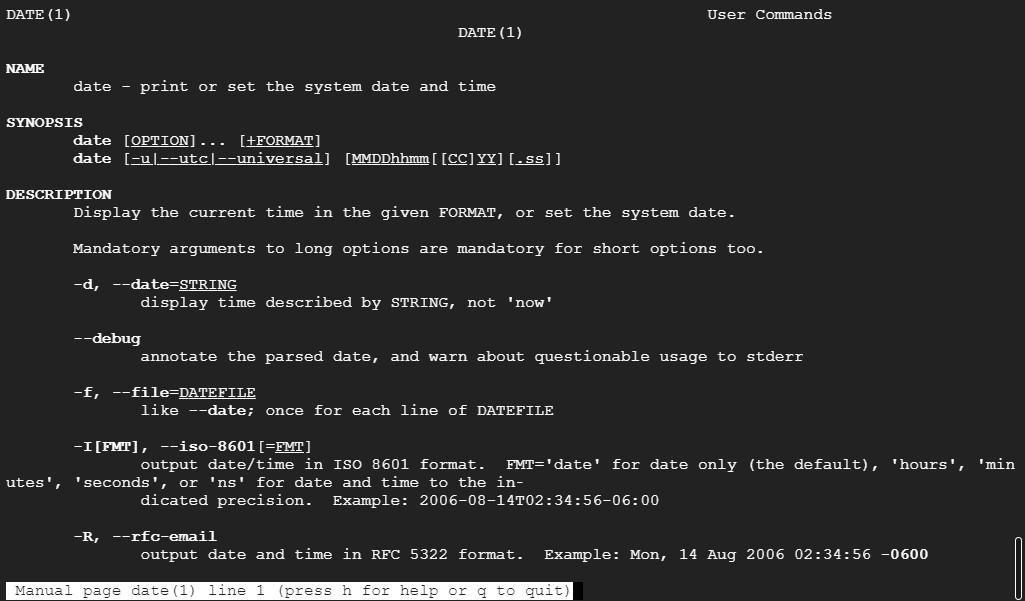
1. **Uname -a**



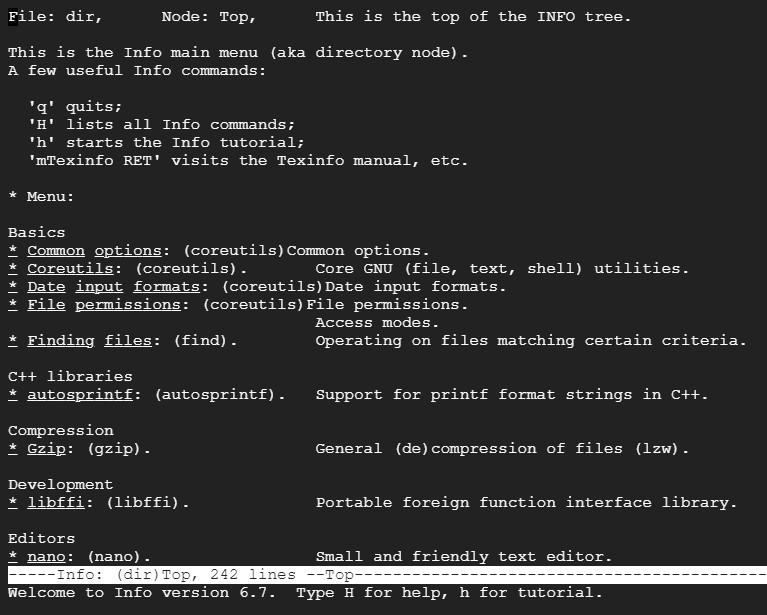
1. **Top**



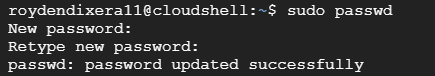
1. **Man**



1. **Info**



1. **Passwd**



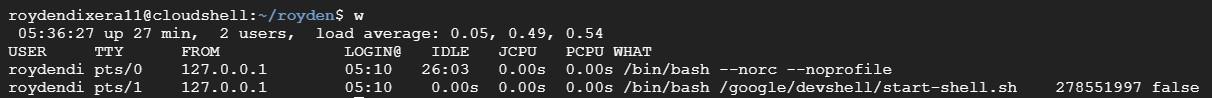
1. **Whatis**



1. **Date**



1. **W**



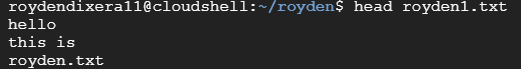
1. **Exit**



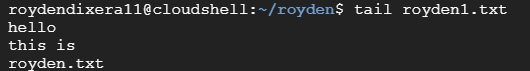
1. **Shutdown**



1. **Head**



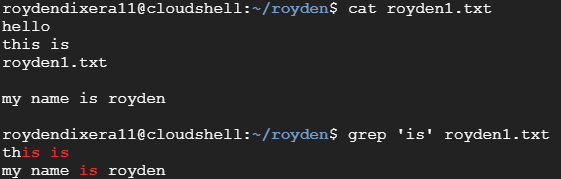
1. **Tail**



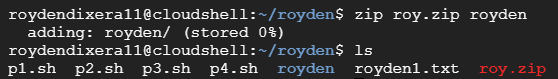
1. **Echo**



1. **Grep**



1. **Zip**



1. **Unzip**



1. **<command name> -h or <command name> –help**

