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

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

**Laboratory Exercise:**

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

1. **Introduction:**

DevOps is a practice that merges development (Dev) and operations (Ops) to enhance the software development lifecycle. By fostering collaboration and automating processes, DevOps aims to shorten development cycles, increase deployment frequencies, and deliver reliable applications. It breaks down silos between development and operations teams, improving communication, collaboration, and productivity through automation of infrastructure, workflows, and continuous performance monitoring.

DevOps fosters a culture of shared responsibility and teamwork. This approach enhances efficiency and productivity by automating repetitive tasks and processes. DevOps also focuses on continuous integration, continuous delivery (CI/CD), and continuous monitoring, ensuring that software is always in a deployable state. Through these practices, organizations can rapidly adapt to market changes, improve customer satisfaction, and maintain a competitive edge.

1. **Need:**

In today's fast-paced digital landscape, businesses require faster software delivery, rapid response to customer feedback, and continuous integration of new features to stay competitive. Traditional methods of software development and operations are no longer sufficient to meet these demands, making DevOps essential. DevOps bridges the gap between development and operations teams, fostering collaboration, streamlining processes, and enabling automation, which are crucial for modern software development.

1. **Faster Delivery:** DevOps accelerates the development, testing, and deployment processes, allowing businesses to release updates and new features more frequently.
2. **Improved Quality**: Automated testing and continuous integration reduce human error, ensuring higher quality software with fewer bugs.
3. **Enhanced Collaboration**: By breaking down silos between teams, DevOps promotes a culture of shared responsibility and teamwork.
4. **Increased Efficiency**: Automation of repetitive tasks saves time and resources, enabling teams to focus on innovation and problem-solving.
5. **Organizational Agility**: DevOps practices enhance the ability to adapt to changing market demands and customer needs, making businesses more agile and responsive.
6. **Usage:**

DevOps is used extensively across various industries to streamline and optimize software development and IT operations. By integrating development and operations, DevOps facilitates continuous delivery and deployment, ensuring that software is always in a releasable state. This approach enables organizations to deliver new features, bug fixes, and updates more rapidly and reliably. Here are some key areas where DevOps is commonly applied:

1. **Continuous Integration and Continuous Deployment (CI/CD):** Automating the integration and deployment processes to ensure that code changes are continuously tested and deployed to production without manual intervention.
2. **Infrastructure as Code (IaC):** Managing and provisioning computing infrastructure through machine-readable configuration files rather than physical hardware configuration or interactive configuration tools.
3. **Automated Testing**: Implementing automated tests to ensure code quality and functionality, reducing the likelihood of defects reaching production.
4. **Monitoring and Logging**: Continuously monitoring applications and infrastructure for performance and errors, and logging relevant data to quickly identify and address issues.
5. **Collaboration and Communication:** Using tools and practices that promote better communication and collaboration between development, operations, and other stakeholders, ensuring everyone is aligned and informed throughout the development lifecycle.
6. **DevOps Lifecycle**

The DevOps lifecycle encompasses various stages that aim to improve the development and operations processes through continuous integration, delivery, and collaboration. Here's an overview of the key stages in the DevOps lifecycle:

1. **Continuous Development:** This stage involves planning and coding. Developers create and manage the software application’s code and features, often using version control systems like Git to track changes and collaborate effectively.
2. **Continuous Integration**: Code changes from multiple contributors are integrated into a shared repository frequently. Automated build and unit testing are triggered with each change, ensuring that the codebase remains stable and any integration issues are identified early.
3. **Continuous Testing**: Automated tests are executed continuously to verify the quality and functionality of the code.
4. **Continuous Deployment**: Code changes that pass all tests are automatically deployed to the production environment. This stage ensures that new features and fixes are delivered to users quickly and reliably.
5. **Continuous Monitoring**: Applications and infrastructure are continuously monitored for performance, errors, and security issues. This stage helps identify and resolve problems in real-time, ensuring the application runs smoothly and efficiently.
6. **Continuous Feedback:** Feedback from users and stakeholders is gathered and analyzed to inform future development and improvements. This stage ensures that the development process is aligned with user needs and business goals.

Each of these stages is interconnected, promoting a cycle of continuous improvement and collaboration between development and operations teams. This approach enables faster delivery of high-quality software, reduces downtime, and enhances the overall user experience.

1. **Advantages and disadvantages:**

#### Advantages:

1. **Increased Deployment Frequency:** Faster and more reliable software releases.
2. **Improved Collaboration**: Better communication between development and operations teams.
3. **Enhanced Product Quality**: Early bug detection through continuous testing.
4. **Greater Efficiency**: Automation reduces time and errors, allowing focus on strategic tasks.
5. **Scalability and Reliability**: Consistent and resilient infrastructure through automated provisioning.

**Disadvantages:**

1. **Cultural Shift**: Requires significant change and buy-in from all stakeholders.
2. **Complexity**: Managing new tools and practices can be challenging.
3. **Initial Investment**: Setup costs for tools, training, and infrastructure.
4. **Security Risks**: Frequent deployments can introduce vulnerabilities if not managed well.
5. **Dependency on Automation**: Heavy reliance on automated processes can cause disruptions if they fail.

Overall, DevOps offers significant benefits, but its successful implementation

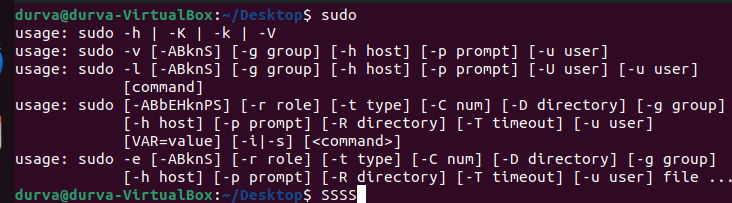
needs careful planning and commitment.

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

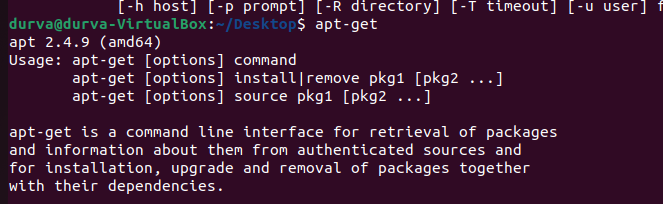
**1.sudo**: sudo (SuperUser DO) Linux command allows you to run programs or other

commands with administrative privileges, just like “Run as administrator”

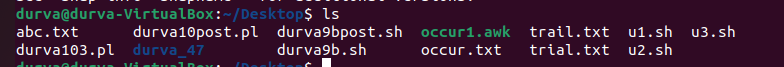
in Windows.



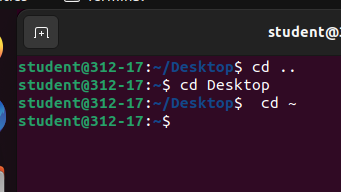
**2. apt-get:** apt-get is one of the most important Ubuntu commands. It is used to install, update, upgrade and remove any package. apt-get basically works on a database of available packages.



**3. ls:** ls (list) command lists all files and folders in your current working directory. You can also specify paths to other directories if you want to view their contents.



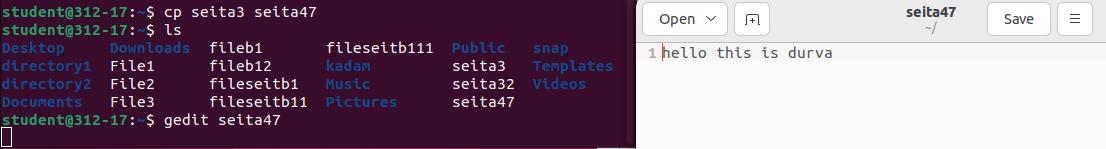
**4.cd:** cd (change director”) Linux command also known as chdir used to change the current working directory. It’s one of the most used basic Ubuntu commands. Using this command is easy, just type cd followed by the the folder name.



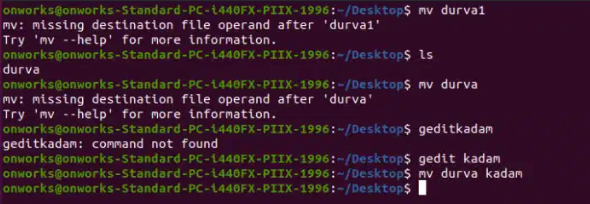
**5.pwd:** pwd (print working directory) Ubuntu command displays the full pathname of the current working directory.



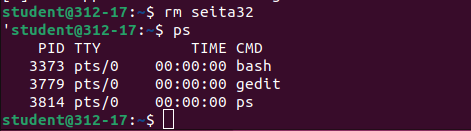
**6.cp:** cp (copy) Linux command allows you to copy a file. You should specify both the file you want to be copied and the location you want it copied to.



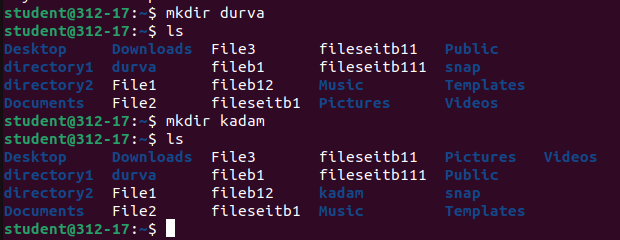
**7.mv:** mv (move) command allows you to move files. You can also rename files by moving them to the directory they are currently in, but under a new name.



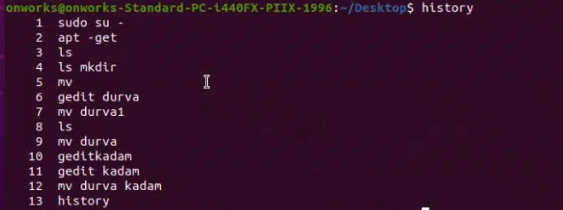
**8. rm:** rm (remove) command removes the specified file.



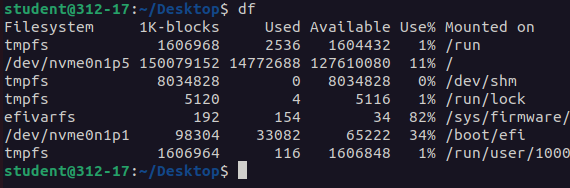
**9.mkdir:** mkdir (make directory) command allows you to create a new directory. You can specify where you want the directory created – if you do not do so, it will be created in your current working directory.



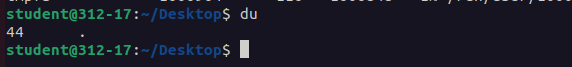
**10.history:** history command displays all of your previous commands up to the history limit.



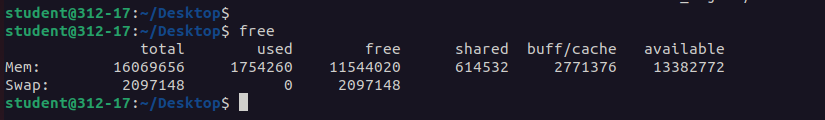
**11.df:** df (display filesystem) command displays information about the disk space usage of all mounted filesystems.



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



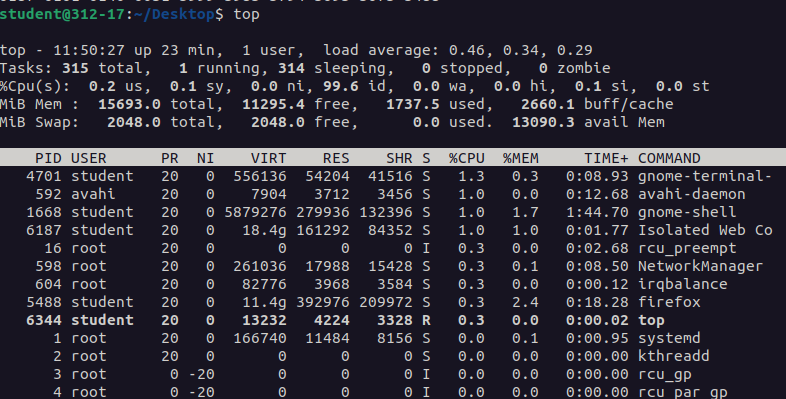
**13**.**free:**  free – Displays the amount of free space available on the system.

****

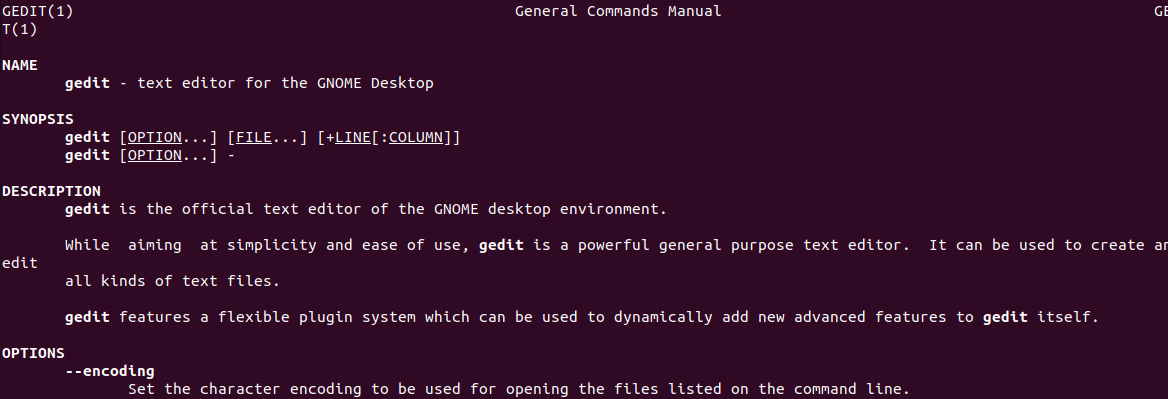
**14.uname -a:** uname -a – Provides a wide range of basic information about the system.



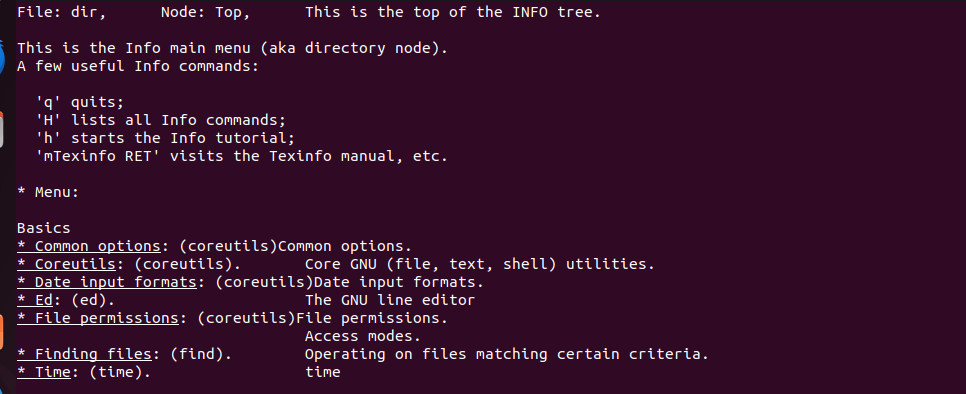
**15.top:** top – Displays the processes using the most system resources at any given time. “q” can be used to exit.



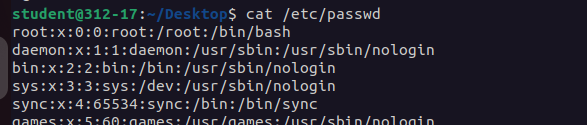
**16.man:** man command displays a “manual page”. Manual pages are usually very detailed.

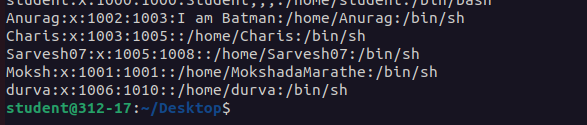


**17.info:** It is Similar to man, but often provides more detailed or precise information.

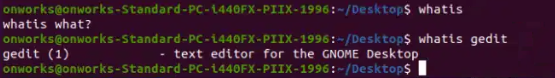


**18.passwd:** passwd Ubuntu basic command is used to change user password using Terminal.





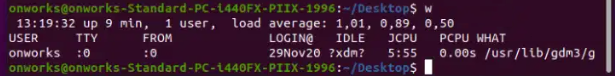
**19. whatis:** whatis command shows a brief description of what is the functionality of specific built-in Linux command.



**20.date:** The simple “date” command displays the current date and time (including the day of the week, month, time, time zone, year).



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



**22.exit:** The command “exit” as the name says it is used to exit from the system

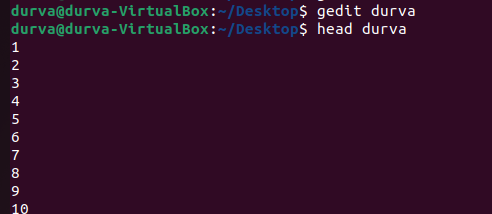
and log out from the current user.

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



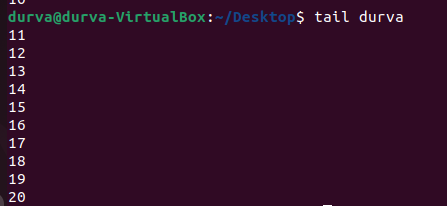
**24.head:** The command “head” prints the top N rows of data of the given input

or file. By default, it prints the first 10 lines of the specified files.

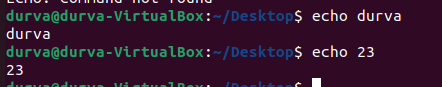
****

**25.tail:** The command “tail” prints the last N rows of data of the given input

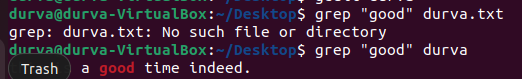
or file. By default, it prints the last 10 lines of the specified files.

****

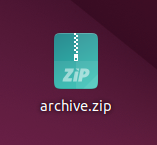
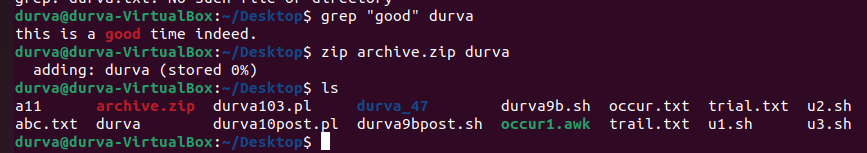
**26.echo**: The command “echo” used to display any expression that is passed as an argument.



**27.grep**: The command “grep” is used to search for a text in the specified file/folder.

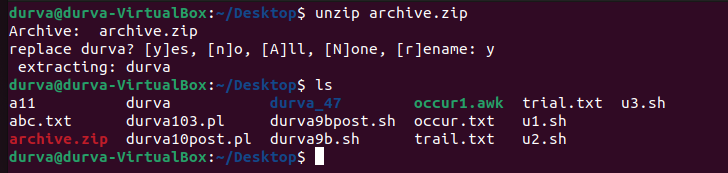


**28.zip:** The command “zip” is used to compress one or more files and store them in a new file with .zip extension.

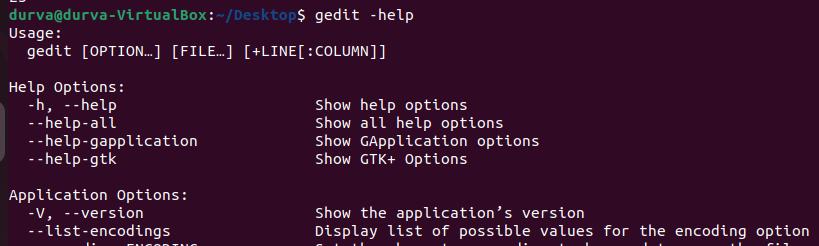


**29.unzip:** The command “unzip” is used to decompress a .zip file and extract

all the files within to current directory.



**30.command name –help:** Provides information about the input command.

****