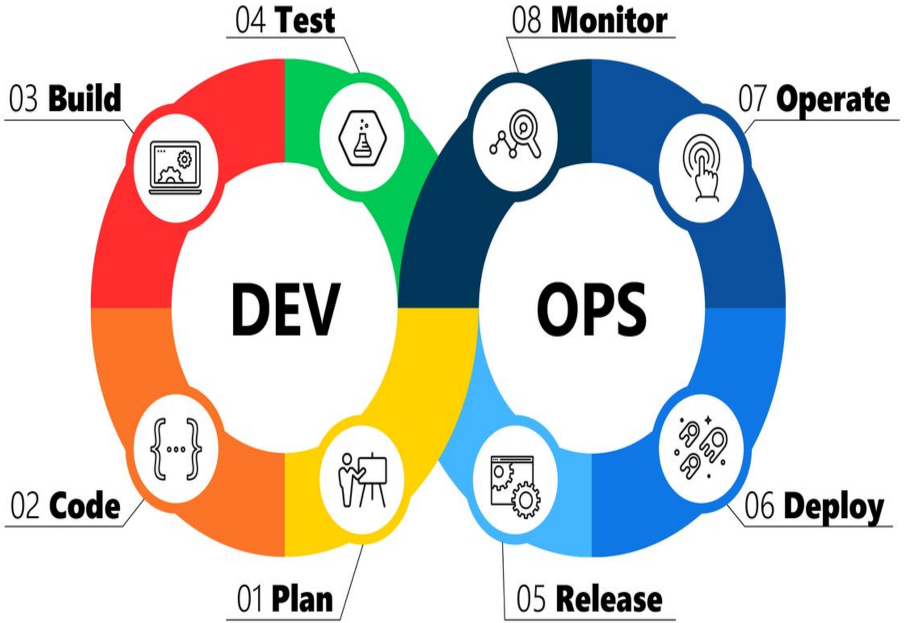
INTRODUCTION TO DEVOPS

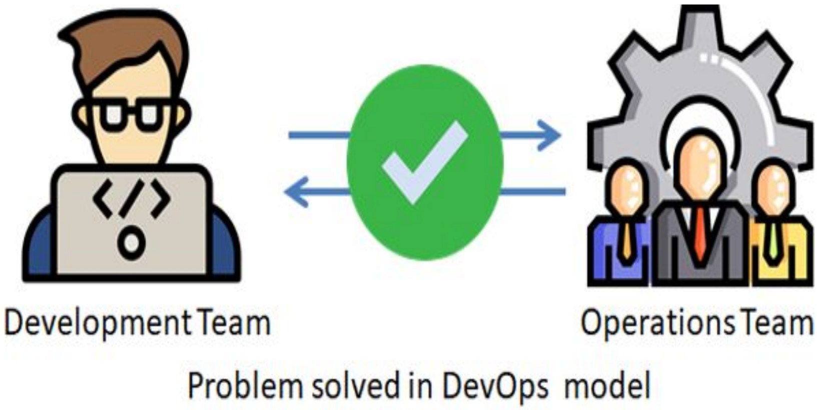
WHAT IS DEVOPS?

- The word DevOps is a combination of two words Development and Operations.
- The **development team** is responsible for developing, designing, and building the application.
- The **operation team** deals with the deployment and testing of the application.

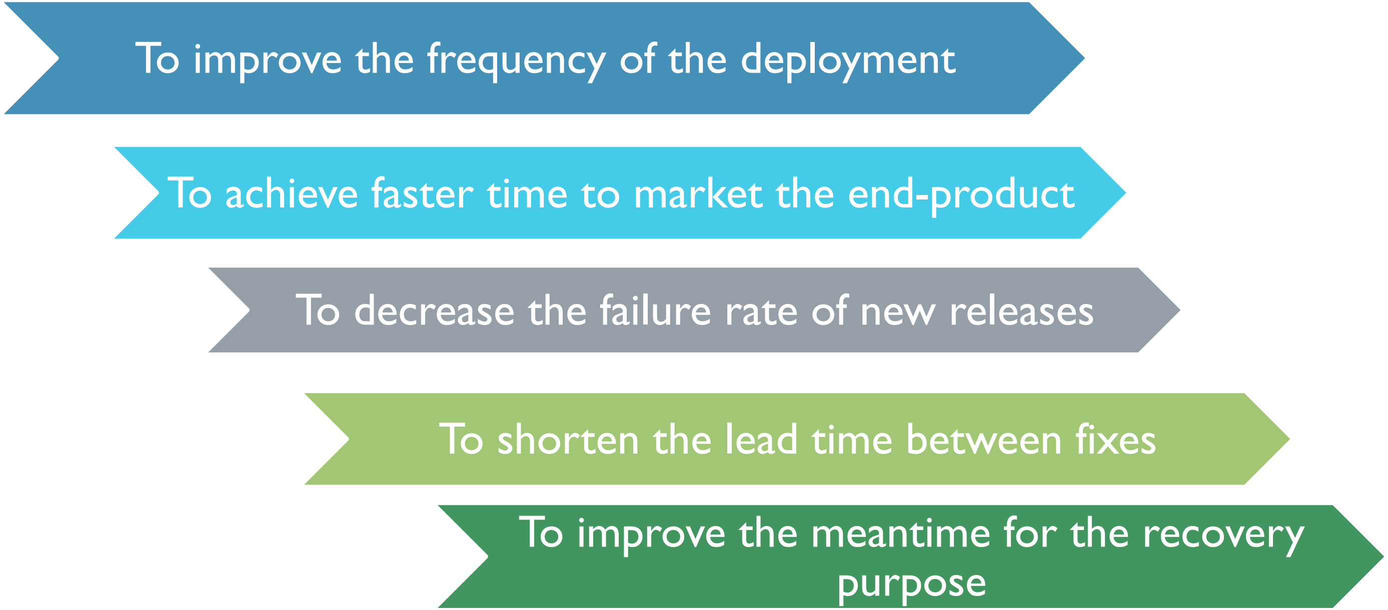


WHY DEVOPS?

- Traditional IT process is time consuming.
- There is a problem of a one-way stream of work, due to which if there is any mistake, the whole process repeats.
- No interaction with operation team



DEVOPS OBJECTIVES



To improve the frequency of the deployment

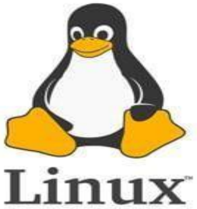
To achieve faster time to market the end-product

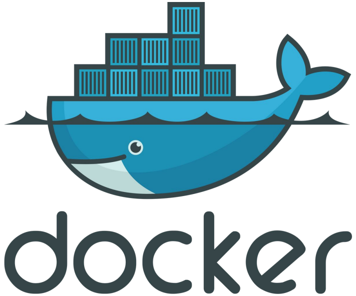
To decrease the failure rate of new releases

To shorten the lead time between fixes

To improve the meantime for the recovery purpose

DEVOPS TOOLS





DEVOPS ADVANTAGES

Time taken to create and deliver software is reduced

Complexity of maintaining an application is reduced

Improved collaboration between developers and operations team

Continuous integration and delivery ensure faster time to market

BASIC PRINCIPALS OF LINUX

BASIC PRINCIPLES OF LINUX

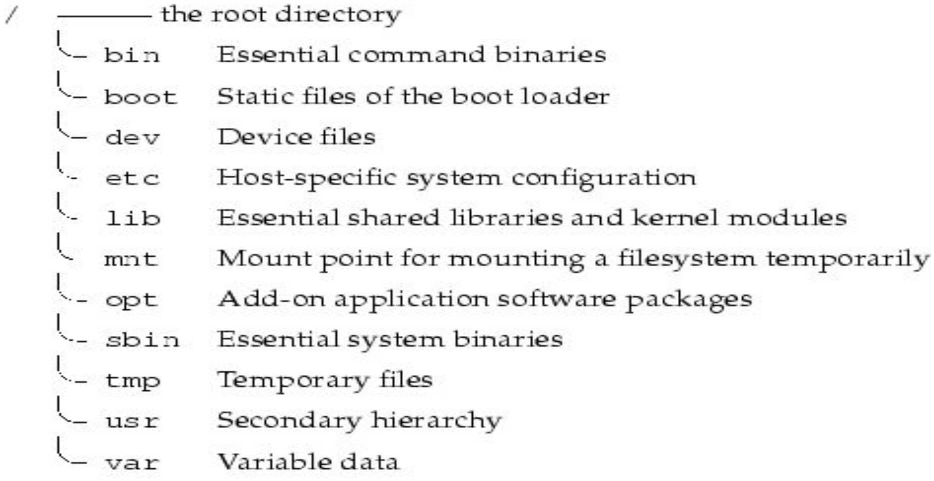
- Everything is a file
- Small, single purpose programs
- Ability to chain programs to perform complex tasks
- Avoid captive user interface
- Configuration data is stored in text
- Linux is case-sensitive

LINUX FILE SYSTEM

- Every file in Linux can be one of the following four types:
  - Ordinary File (contain data, info)
  - Directories (hold files & other directories)
  - Devices (for accessing the hardware)
  - Links (Pointer to another file)
    - Hard Link
    - Soft Link

TYPICAL DIRECTORY STRUCTURE

- Linux Filesystem is laid out in a hierarchical tree structure.
- Top level directory is called root "/"



```
/      ———— the root directory
 └─ bin     Essential command binaries
 └─ boot    Static files of the boot loader
 └─ dev     Device files
 └─ etc     Host-specific system configuration
 └─ lib     Essential shared libraries and kernel modules
 └─ mnt     Mount point for mounting a filesystem temporarily
 └─ opt     Add-on application software packages
 └─ sbin    Essential system binaries
 └─ tmp     Temporary files
 └─ usr     Secondary hierarchy
 └─ var     Variable data
```

BASIC LINUX COMMANDS

- **cd**, change directory
  - cd ..
  - cd –
  - cd ~/mydir
  - cd /home/usman
  - cd
- **su**, switch user, su – (complete user environment)
- **id**, print user and group ids
-

BASIC LINUX COMMANDS

- **man**, manual pages
  - man <command>
- **info**, information pages
  - info <command>
- **command --help**, basic help by author
  - ls --help, man --help
- **pwd**, present working directory

BASIC LINUX COMMANDS

- **ps**, process information
  - ps aux
  - pstree
  - ps fax
  - top
- **top**, top (table of processes)
- **free**, memory information
  - free –m
- **head**, output the first part of files
  - head [-n <lines>] <filename>
- **tail**, output the last part of files
  - tail [-n <lines>] <filename>

BASIC LINUX COMMANDS

- **cat**, concatenate/display files
  - cat /home/usman/myfile
- **clear**, clears the screen
- **date**, see/modify system date & time
  - date, date [MMDDhhmm[[CC]YY][.ss]]
- **df**, disk space usage
  - df –h
- **du**, file space usage
  - du –sh
- **uname**, print system info
  - uname [-a, -s, -n, -r, -v, -m]

BASIC LINUX COMMANDS

- **cp**, copy files and directories
  - cp [options] *file* *destination*
- More than one file may be copied at a time if the destination is a directory:
  - cp [options] *file1* *file2* *destination*
- If the destination is a directory, the copy is placed there
- If the destination is a file, the copy overwrites the destination
- If the destination does not exist, the copy is renamed

BASIC LINUX COMMANDS

- **mv**, move and/or rename files and directories
  - mv [options] *file* *destination*
- More than one file may be moved at a time if the destination is a directory:
  - mv [options] *file1* *file2* *destination*
- In **mv** also, the destination works like **cp**
- **mkdir**, creates directories
  - mkdir <directory name/path>
- **rmdir**, removes empty directories
  - rmdir < directory name/path>

BASIC LINUX COMMANDS

- **touch**, create empty files or update file timestamps
  - touch <file name/path>
- **rm**, remove files
  - rm [options] <file name/path>
  - **rm -i *file*** (interactive)
  - **rm -r *directory*** (recursive)
  - **rm -f *file*** (force)
- **rm –r**, recursively removes directory trees
  - rm –rf <directory name/path>

BASIC LINUX COMMANDS

- find, finding files and directories
  - find <path> –iname <file name>
  - find <path> -size +100
  - find <path> -user usman –o –group it
  - find <path> -perm 755

FILE PERMISSIONS IN LINUX

Octal	Binary	Permissions
0	000	_ _ _
1	001	_ _ x
2	010	_ w _
3	011	_ w x
4	100	r _ _
5	101	r _ x
6	110	r w _
7	111	r w x

BASIC LINUX COMMANDS

- **chmod**, changing permissions of files/directories
  - Octal mode
    - chmod 644 <file name/path>
    - chmod 755 <file name/path>
    - chmod 6 <file name/path>
  - Symbolic mode
    - chmod u+x, g-r, o+x <file name/path>
    - chmod a+x <file name/path>
    - chmod =x <file name/path>
- **chown,** changing ownership of files/directories
- **chgrp**, changing group ownership of files/directories

BASIC LINUX COMMANDS

- **stat**, display the file status
  - stat <file name/path>
- ssh, Open SSH client for remote login
  - ssh <username>@<hostname>
  - ssh –l <username> <hostname>
  - ssh <hostname>
- **scp**, secure copy (remote file copy)
  - scp <file name> <username>@<hostname>:<path>
  - scp <username>@<hostname>:<path> <local path>

THANK YOU