

## Prerequisite- Writeup

**Writeup name:** Introduction to Advanced DevOps.

**Aim:** To go over the prerequisite to Advanced DevOps.

**Theory:**

- 1. Can a computer system run without an operating system, If yes why, If no why!!! State your views.**

**Ans:**

- Yes a computer can definitely work without an OS. But it won't be able to perform the same complex task our computers are used to perform (such as playing music, games, document preparation etc) as smoothly and efficiently. It is an OS that is providing an environment for the process and applications to run smoothly without the user being bothered about what is happening at the backend your computer. Without an operating system your laptop is just a box of bits that do not know how to communicate with one another, or you.
- Without an operating system using and enforcing a standard, systematic approach to running the computer, you're put in the position of writing code (or programs) that must tell the computer exactly what to do. So, if you want to type up a document in a word processing program, you'd have to create from scratch code that tells your computer to respond to each character pressed on your keyboard. Then you'd have to write a code that told the computer how those responses must translate to a screen. You'd have to tell your computer how to draw the character you want! Think of every single option or possibility your word processing program has. You'd have to write code for every single one of those directly onto your hard drive.
- So we can conclude that without an OS the computer will function with low level task, but it necessary to function and perform the tasks easily which we are able to perform with the computer in the real world.

- 2. What is the importance of kernel in a Operating system, will an OS function without a kernel!!!**

**Ans:**

- Kernel is central component of an operating system that manages operations of computer and hardware. It basically manages operations of memory and CPU time. It is core component of an operating system. Kernel acts as a bridge between applications and data processing performed at hardware level using inter-process communication and system calls.
- It is the computer program that first loaded on start-up the system (After the bootloader). Once it is loaded, it manages the remaining start-ups. It also manages memory, peripheral, and I/O requests from software.
- Moreover, it translates all I/O requests into data processing instructions for the CPU. It manages other tasks also such as memory management, task management, and disk management.
- NO, it can never work without a kernel. A kernel is a small piece of code which embeds

in the hardware and this is the only component which make any communication with hardware possible. If this wouldn't be present you will lose any communication with hardware. And since you need a platform to run, your software will not detect any and will never work.

### 3. Why is Linux Operating system and its various Distros dominating the Tech sphere as compared to its competing OS (Mac OS, windows etc.)

**Ans:**

- Linux is an operating system -- very much like UNIX -- that has become very popular over the last several years. It is free and runs on PC platforms, it gained a sizeable audience among hard-core developers very quickly. Linux has a dedicated following and appeals to several different kinds of people.
- Following are some of the reasons why Linux is so popular –
  - Open Source, use can customize the OS according to its needs.
  - It is highly secure, many of the servers of big companies and tech giants use Linux in their server due to its high security and reliability.
  - Powerful commands and utilities, Linux comes with huge variety of network and other commands and utilities which helps the user in managing and monitoring OS and networks.
  - It supports many different architectures.
  - It can give best performance even on the lowest end hardware configuration. On the other hand, Windows and MacOS need high end hardware configuration for better performance.
  - It has many distributions which users can select according to his/her needs.

Hence, we can say that Linux dominates the Tech sphere.

### 4. Jot down at least 10 Linux commands and explain them.

**Ans:**

1. **pwd:** It writes the full pathname of the current working directory to the standard output.

```
abhi@abhi-VirtualBox:~$ pwd
/home/abhi
```

2. **ls:** Used to list all files.

```
abhi@abhi-VirtualBox:~$ ls
armstrong.sh  exp3tcp.tcl      factwhile      prime.sh      sithro.awk
demo          exp3tcpudp.tcl   factwhile.sh   Public        snap
demo1.tcl     exp3udp.tcl      hello.sh       s1.awk        sum.sh
demo.tcl      exp4tcpudp.tcl   inputdata.sh   s1.nam        TCP.nam
Desktop       exp5s1pdr.awk    Music          s1pdr.awk     TCP.tr
Documents     exp5sim1pdr.awk  os1.txt        sithro.awk    Templates
Downloads     exp5sim1thro.awk os2.txt        s1.tr         Videos
evenodd.sh    fac.sh           Pictures       sim1.nam
exp3.tcl      fact.sh          Prac1          sim1.tr
```

3. **mkdir:** The mkdir command in Linux allows users to create or make new directories. mkdir stands for "make directory."

```
abhi@abhi-VirtualBox:~$ mkdir Prac1
abhi@abhi-VirtualBox:~$
```

4. **ping (Packet Internet Groper):** ping command sends packet requests to the address you specify to test the connectivity between 2 nodes. Ping command sends an ICMP ECHO\_REQUEST packet to the target host and waits to see if it replies.

```
abhi@abhi-VirtualBox:~$ ping google.com
PING google.com (172.217.160.206) 56(84) bytes of data.
64 bytes from bom07s16-in-f14.1e100.net (172.217.160.206): icmp_seq=1 ttl=119 time=1210 ms
64 bytes from bom07s16-in-f14.1e100.net (172.217.160.206): icmp_seq=3 ttl=119 time=18.5 ms
64 bytes from bom07s16-in-f14.1e100.net (172.217.160.206): icmp_seq=4 ttl=119 time=16.9 ms
```

5. **echo:** The command echo is one of the most basic and frequently used commands in Linux. The arguments passed to echo are printed to the standard output.

```
abhi@abhi-VirtualBox:~$ echo "Abhi"
Abhi
abhi@abhi-VirtualBox:~$ echo -e "Abhishek \n Yadav"
Abhishek
Yadav
abhi@abhi-VirtualBox:~$ echo -e "Abhishek \t Yadav"
Abhishek      Yadav
abhi@abhi-VirtualBox:~$ echo -e "Abhishek \v Yadav"
Abhishek
Yadav
```

6. **history:** It is used to view the previously executed command.

```
abhi@abhi-VirtualBox:~$ history
1  sudo apt install build-essential dkms linux-headers-$(uname-r)
2  ns exp4tcpudp.tcl
3  ns demo.tcl
4  ns demo1.tcl
5  sudo apt-get update -y
6  sudo apt-get install -y leafpad
7  snap install leafpad --classic
8  echo "Abhi"
9  echo "Abhi" "Abhishek \t Yadav"
10 echo "Abhi"
11 echo -e "Abhishek \t Yadav"
12 echo -e "Abhishek \n Yadav"
13 echo -e "Abhishek \v Yadav \v Navi Mumbai"
14 clear
15 mkdir Prac1
16 clear
17 data
18 date
19 date -u
20 cal
```

7. **traceroute:** Traceroute prints the route that packets take to a network host. Traceroute utility uses the TTL field in the IP header to achieve its operation.

```
abhi@abhi-VirtualBox:~$ traceroute google.com
traceroute to google.com (142.250.76.206), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.297 ms  0.221 ms  0.665 ms
 2  10.0.2.2 (10.0.2.2)  669.209 ms  668.864 ms  668.853 ms
abhi@abhi-VirtualBox:~$ traceroute youtube.com
traceroute to youtube.com (172.217.27.206), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.349 ms  0.281 ms  0.262 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * 10.0.2.2 (10.0.2.2)  704.653 ms  704.551 ms
```

8. **hostname:** hostname is to identify in a network. Execute hostname command to see the hostname of your box.

```
abhi@abhi-VirtualBox:~$ hostname
abhi-VirtualBox
```

9. **top:** top command displays processor activity and also displays tasks managed by kernel in real-time. It'll show processor and memory are being used and other information like running processes.

```
abhi@abhi-VirtualBox:~$ top
top - 01:34:56 up 6 min, 1 user, load average: 0.14, 0.18, 0.10
Tasks: 177 total, 2 running, 175 sleeping, 0 stopped, 0 zombie
%Cpu(s): 8.6 us, 0.3 sy, 0.0 ni, 91.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 4659.9 total, 3165.5 free, 621.1 used, 873.3 buff/cache
MiB Swap: 448.5 total, 448.5 free, 0.0 used, 3797.9 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 1527 abhi       20   0 3452144 339464 123864 S   5.3   7.1   0:10.21 gnome-shell
 1314 abhi       20   0 538852 67416 44424 S   3.3   1.4   0:04.21 Xorg
 1874 abhi       20   0 822444 50736 38948 S   1.7   1.1   0:01.77 gnome-terminal-
 239  root       20   0      0      0      0 I   0.3   0.0   0:00.20 kworker/0:4-events
 245  root      -51   0      0      0      0 S   0.3   0.0   0:00.10 irq/18-vmwgfx
 1986 abhi       20   0 20504 3664 3128 R   0.3   0.1   0:00.05 top
    1  root       20   0 102312 11504 8296 S   0.0   0.2   0:01.28 systemd
    2  root       20   0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3  root        0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4  root        0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6  root        0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-kblockd
    8  root       20   0      0      0      0 I   0.0   0.0   0:00.06 kworker/u2:0-events_unbound
    9  root        0 -20      0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
   10  root       20   0      0      0      0 S   0.0   0.0   0:00.10 ksoftirqd/0
   11  root       20   0      0      0      0 R   0.0   0.0   0:00.47 rcu_sched
   12  root        rt   0      0      0      0 S   0.0   0.0   0:00.00 migration/0
   13  root      -51   0      0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
   14  root       20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   15  root       20   0      0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
   16  root        0 -20      0      0      0 I   0.0   0.0   0:00.00 netns
   17  root       20   0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_kthre
   18  root       20   0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
   19  root       20   0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
   20  root       20   0      0      0      0 S   0.0   0.0   0:00.00 kauditd
   21  root       20   0      0      0      0 S   0.0   0.0   0:00.00 khungtaskd
   22  root       20   0      0      0      0 S   0.0   0.0   0:00.00 oom_reaper
```

10. **cat:** It allows us to create single or multiple files, view contain of file, concatenate files and redirect output in terminal or files.

```
abhi@abhi-VirtualBox:~$ cat > os1.txt
hello world
abhi
^C
abhi@abhi-VirtualBox:~$ cat os1.txt
hello world
abhi
abhi@abhi-VirtualBox:~$ cat > os2.txt
unix
^C
abhi@abhi-VirtualBox:~$ cat os2.txt
unix
abhi@abhi-VirtualBox:~$ cat os1.txt;cat os2.txt
hello world
abhi
unix
abhi@abhi-VirtualBox:~$ cat os1.txt > os2.txt
abhi@abhi-VirtualBox:~$ cat os2.txt
hello world
abhi
```

**5. Can the established network industry function without the OSI model? If so, state your views regarding it, also specify its importance.**

**Ans:**

- No, the network industry cannot function without OSI model.
  - OSI (Open Systems Interconnection) is the model of reference for understanding relationships is a conceptual framework. It aims to guide vendors and developers so they can interoperate with the digital communication products and software programs they create and to provide a clear framework describing the functions of a network or telecommunications system.
  - The principal concept of OSI is that the communication process between two endpoints can be separated into seven different groups of associated functions or layers in a network. Each communicating user or program is on a device with seven function layers.
  - Every layer serves above it in this architecture and the layer beneath it is served in turn. Thus, there will be a data flow through the layers of the source computer and across the network and through the layers at the receiving computer in a given message between users.
6. What is the most important hardware component required to ensure you could connect to the internet (Is it the modem/router)

**Ans:**

- Between Modem and Router, Modem is the most important hardware component, which ensures that we are connected to the internet. We can connect our computer directly to our modem using an Ethernet cable. Also, many modems include a built-in router, so we have the option of creating a Wi-Fi network without buying extra hardware.
- A modem transmits, receives and converts data. Whether you use DSL, cable, fiber or satellite internet, a modem is the device that translates signals from its digital or analog form to what you see on your screen. In other words, a modem gets the internet to your devices. A modem can function independently from a router.

**7. What is software development life cycle (SDLC)? Do we as software engineers actually require a SDLC, state your views.**

**Ans:**

- SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.
- The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software.
- Following are the phases in SDLC cycle.
  - Requirement analysis
  - Planning
  - Software design such as architectural design
  - Software development
  - Testing
  - Deployment
- It is important to have an SDLC in place as it helps to transform the idea of a project into a functional and completely operational structure. In addition to covering the technical aspects of system development, SDLC helps with process development, change management, user experience, policies, project planning with respect to resource allocation and release of deliverables at the specified scheduled deadlines .

**8. What is cloud computing?**

**Ans:**

- Cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet (“the cloud”) to offer faster innovation, flexible resources, and economies of scale. You typically pay only for cloud services you use, helping lower your operating costs, run your infrastructure more efficiently and scale as your business needs change.
- Cloud computing works by allowing client devices to access data over the internet, from remote servers, databases and computers.
- An internet network connection links the front end (includes the accessing client device, browser, network and cloud software applications) with the back end, which consists of databases, servers and computers).
- Communications between the front and back ends are managed by a central server.
- Examples of cloud computing –
  - Google Docs, Microsoft Office 365.
  - Email, Calendar, Skype, WhatsApp.
  - Zoom
  - AWS Lambda
- A cloud service has three distinct characteristics

- Users can access large amounts of computing power on demand. It is typically sold by the minute or the hour.
  - It is elastic, a user can have as much or as little of a service as they want at any given time.
  - The service is fully managed by the provider (the consumer needs nothing but a personal computer and internet access).
9. What is DevOps Ecosystem? How is DevOps different from your traditional software development?

**Ans:**

- DevOps is currently the most popular methodology for **developing enterprise scale software systems**. It is essentially methodology that emphasizes communication, collaboration and integration between software developers and personnel involved in operations such as IT administration. DevOps was motivated by the need to address the implications of the well-known interdependence between software development and IT operations. DevOps is already helping organizations to rapidly produce software products and services.
- DevOps organizations make the release of software into a non-event much as possible. They reduce risk by daily integrating code into the trunk, automating testing, ensuring all environments are in sync, reducing batch sizes (as mentioned above) etc. Differential points are mentioned as follow:

#### **Batch Sizes: Go from Big to Micro:**

- Traditional IT has a bias for going big and for good reason. First, most development shops grew out of the waterfall method which by its inherent nature takes a lot of time. A DevOps organization takes the opposite point of view and believe that small is beautiful. They understand that large batch sizes are inherently complex, risky (since there are so many moving parts), and hard to coordinate.

#### **Organization: From Skill Centric Silos to Dedicated Cells:**

- Traditional IT is organized around skill centric silos. For the most part silos work, they band like skills together, drive greater utilization, and benefit from economies of scale. However, where these “cost optimized” silos break down is at the hand offs. A DevOps organization also operates in a silo but at a different cross section. Here teams are arranged in “cells”, consisting of dedicated cross functional teams, and focused on only one application.

#### **Scheduling: Centralize to Decentralize & Continuous:**

- Efficient scheduling is at the heart of a Traditional IT organization. Since resources are pooled, projects are usually clamoring for access to SMEs and / or infrastructure. In a DevOps organization scheduling is pushed to the local cell level. The combination of smaller batch sizes, dedicated teams, and automated processes makes scheduling simpler to operate.

#### **Release: Turn a High-Risk Event to a Non-Event:**

- In a Traditional IT organization, releasing software into production is a high-risk proposition. It is fraught with issues, escalations, and constant fire-fighting. DevOps organizations, on the other hand, make the release of software into a non-event much as possible. They reduce risk by daily integrating code into the trunk, automating testing, ensuring all environments are in sync, reducing batch sizes (as mentioned above) etc.

10. How is cloud computing and DevOps interlinked to form an Advanced DevOps category?

**Ans:**

- DevOps has been responsible for providing a new and wider perspective on the delivery of software solutions. The winning combination of DevOps and Cloud Computing is introducing new benefits for cloud development. DevOps is a new movement in the IT sector that involves collaboration between the development engineer and the IT operations team. The prime objective of DevOps is the improvement of communication and coordination among the two crucial teams for application development.
- An evaluation of the close links between DevOps and Cloud Computing can help in clear anticipation of the role of DevOps as a game-changer in cloud development.  
Cloud computing is centralized in nature, thereby providing the perfect foundation to run DevOps automation.
- The cloud offers a centralized platform for testing, deployment, and production. At many instances in the past, the distributed enterprise systems did not align perfectly with centralized software deployment. Therefore, using cloud and DevOps together leads to the resolution of many issues pertaining to distributed complexity.
- The relationship between DevOps and Cloud Computing builds primarily on the following crucial benefits of DevOps.
  1. Provides increased speed in the total application workflow.
  2. Provides enhanced control over the application lifecycle by allowing factors like user feedback as a part of product design.
  3. Enables better collaboration among teams across various specializations.
  4. Thrives totally on customer focus through better control over workflows.
  5. Provides security fixes, a centralized architecture, and compliance measures for ensuring safeguards for sensitive information in the application development workflow.

**Conclusion:** Thus , successfully understood and had a run through the pre-requisites of Advanced DevOps