1. Linux:

1. Provide steps to create a directory inside a directory where the parent directory does not exist.

Ans:- mkdir -p /path/to/non\_existent\_parent/new\_directory

2. How to install a package on a Linux server when there is no internet connection?

Ans:- Install a package on a Linux server without an internet connection by following these steps.

Download the package and its dependencies on a system with internet access.

Transfer the downloaded files to the offline server.

Install the package using the dpkg command for Debian-based systems.

For Exp:- dpkg -i package.deb

3. How to access specific folders of Server A from Server B and Server C?

Ans:- SSH and SCP to access specific folders on Server A from Server B and Server C. Here are the commands to copy files/folders from Server A to Server B:

To copy from Server A to Server B:

scp user@serverA:/path/to/source/file user@serverB:/path/to/destination/

To copy from Server A to Server C:

scp user@serverA:/path/to/source/file user@serverC:/path/to/destination/

4. How to check all the running processes from a server?

Ans:- Using the ps command to list all running processes. To list all processes, you can use:

ps aux

5. Provide the command to delete all the files older than X days inside a specific directory.

Ans:- Using the find command to delete files older than X days. For example, to delete files older than 7 days in a specific directory:

find /path/to/directory -type f -mtime +7 -exec rm {} \;

6. Create a shell script to identify the process ID

a. script should as a user input for process ID

b. If the process exists script should print the process ID and exit

c. If the process doesn't exist script should print the process doesn't exist and asks for another input.

Ans:-

#!/bin/bash

read -p "Enter the process ID: " pid

if ps -p $pid > /dev/null

then

echo "Process $pid exists."

else

echo "Process $pid does not exist."

fi

2. Docker:

1. What is docker and why do we need it?

Ans:- Docker is a containerization platform that allows you to package applications and their dependencies into a standardized unit called a container. Containers are lightweight, portable, and isolated, making it easier to deploy and run applications consistently across different environments.

2. Write a docker file for a sample Java/python application.

Ans:-

FROM openjdk:11 # Use the Java base image

WORKDIR /app

COPY . /app # Copy application code into the container

RUN javac Main.java # Replace with your build commands

CMD ["java", "Main"] # Replace with your application's execution command

3. What is the docker lifecycle?

Ans:- The Docker lifecycle includes building an image from a Dockerfile, creating containers from images, starting and stopping containers, and managing container resources and networks.

4. What is the difference between an image and a container?

Ans:- An image is a lightweight, stand-alone, and executable package that includes the application code, libraries, and dependencies. A container is an instance of an image that can be run, started, stopped, and managed. Containers are the executable units of Docker.

5. How to check docker container logs? Provide the command for the same.

Ans:- To check the logs of a Docker container, i can use the docker logs command followed by the container ID or name. For example:

docker logs <container\_id\_or\_name>

3. Kubernetes:

1. What are different types of services?

Ans:- In Kubernetes, there are different types of services:

ClusterIP: Provides a stable internal IP address for communication within the cluster.

NodePort: Exposes a service on a static port on each node, allowing external access.

LoadBalancer: Exposes the service externally using a cloud provider's load balancer.

ExternalName: Maps a service to an external DNS name.

2. What is a pod?

Ans:- A Pod is the smallest deployable unit in Kubernetes. It can contain one or more containers that share the same network and storage. Pods are used to deploy and manage applications.

3. Create a pod with the above created custom image when a pod dies k8s should automatically restart.

Ans:- Using a Deployment resource to ensure that a pod automatically restarts when it fails. Define the pod and its containers within a Deployment configuration.

4. How to access the custom application with a specific port?

Ans:- By exposing my application to external traffic using a Kubernetes Service with a NodePort or LoadBalancer type, which maps to a specific port on the nodes or uses a cloud provider's load balancer to route traffic to the application.