1. Could you please explain the security mechanisms used by package repositories to prevent the installation of corrupted or tampered packages on a system?

Package repositories use **GPG signatures** to verify package authenticity, **hash checksums** (e.g., SHA256) to detect corruption, and **secure protocols** (e.g., HTTPS) to prevent tampering during download.

2. We are using the following APT source list in `/etc/apt/sources.list`:
   Could you explain the role of each repository component (e.g., main, universe, multiverse, security, updates, back ports)?

```
deb http://ir.archive.ubuntu.com/ubuntu focal main restricted
deb http://ir.archive.ubuntu.com/ubuntu focal-updates main restricted
deb http://ir.archive.ubuntu.com/ubuntu focal universe
deb http://ir.archive.ubuntu.com/ubuntu focal-updates universe
deb http://ir.archive.ubuntu.com/ubuntu focal multiverse
deb http://ir.archive.ubuntu.com/ubuntu focal-updates multiverse
deb http://ir.archive.ubuntu.com/ubuntu focal-backports main restricted
universe multiverse
deb http://security.ubuntu.com/ubuntu focal-security main restricted
deb http://security.ubuntu.com/ubuntu focal-security universe
deb http://security.ubuntu.com/ubuntu focal-security multiverse
```

- **main**: Free, officially supported Ubuntu software.
- **restricted**: Proprietary drivers and firmware.
- **universe**: Community-maintained, open-source software.
- **multiverse**: Non-free or legally restricted software.
- **security**: Critical security updates for all components.
- **updates**: General updates and bug fixes post-release.
- **backports**: Newer software versions not in standard release.

3. What is the difference between KVM and QEMU? Also, what exactly is `qemu-kvm`?

KVM is a Linux kernel module for hardware-accelerated virtualization, while **QEMU** is a user-space emulator for virtual machines. qemu-kvm is QEMU integrated with KVM for faster, hardware-assisted virtualization.

4. What are the differences between a virtual machine and a container? For each of the following scenarios, which one would you recommend and why?

- A database under heavy load with frequent disk writes
    - VM, for better isolation and disk performance.
- A stateless web application
    - Container, for lightweight, scalable deployment.
- A monolithic web application with high resource demands
    - VM, for dedicated resources and isolation.
- An application that requires a specific hardware driver (e.g., a printer driver)
    - VM, for direct hardware access.
- Graphical (GUI) applications
    - VM, for full OS environment with GUI support.
- Applications that depend on multiple system services initialised at boot (e.g., mail servers)
    - VM, for independent service initialization.
- Applications that require custom kernel modules
    - VM, for kernel customization.
- Lightweight programs with minimal resource needs
    - Container, for minimal resource use.
- Applications that need to interact with other processes directly
    - Container, for shared host process access.

5. difference between a hashing algorithm and an encryption algorithm?

A **hashing algorithm** creates a fixed-size, irreversible digest (e.g., SHA256) for data integrity, while an **encryption algorithm** reversibly transforms data (e.g., AES) for confidentiality using a key.

6. I have a directory with many configuration files. How can I generate checksums to detect even a single-character change in any file?

```
Desktop $ ls -lrth x/

total 8

-rw-r--r--@ 1 arash  staff   973B May 16 18:08 main.py

Desktop $ find /Users/arash/Desktop/x/ -type f -exec sha256sum {} \; >
/Users/arash/Desktop/checksums.txt

Desktop $ sha256sum -c checksums.txt
/Users/arash/Desktop/x//main.py: OK

Desktop $ echo "# change" >> /Users/arash/Desktop/x/main.py
```

```
Desktop $ sha256sum -c checksums.txt
sha256sum: checksums.txt: no properly formatted checksum lines found
```