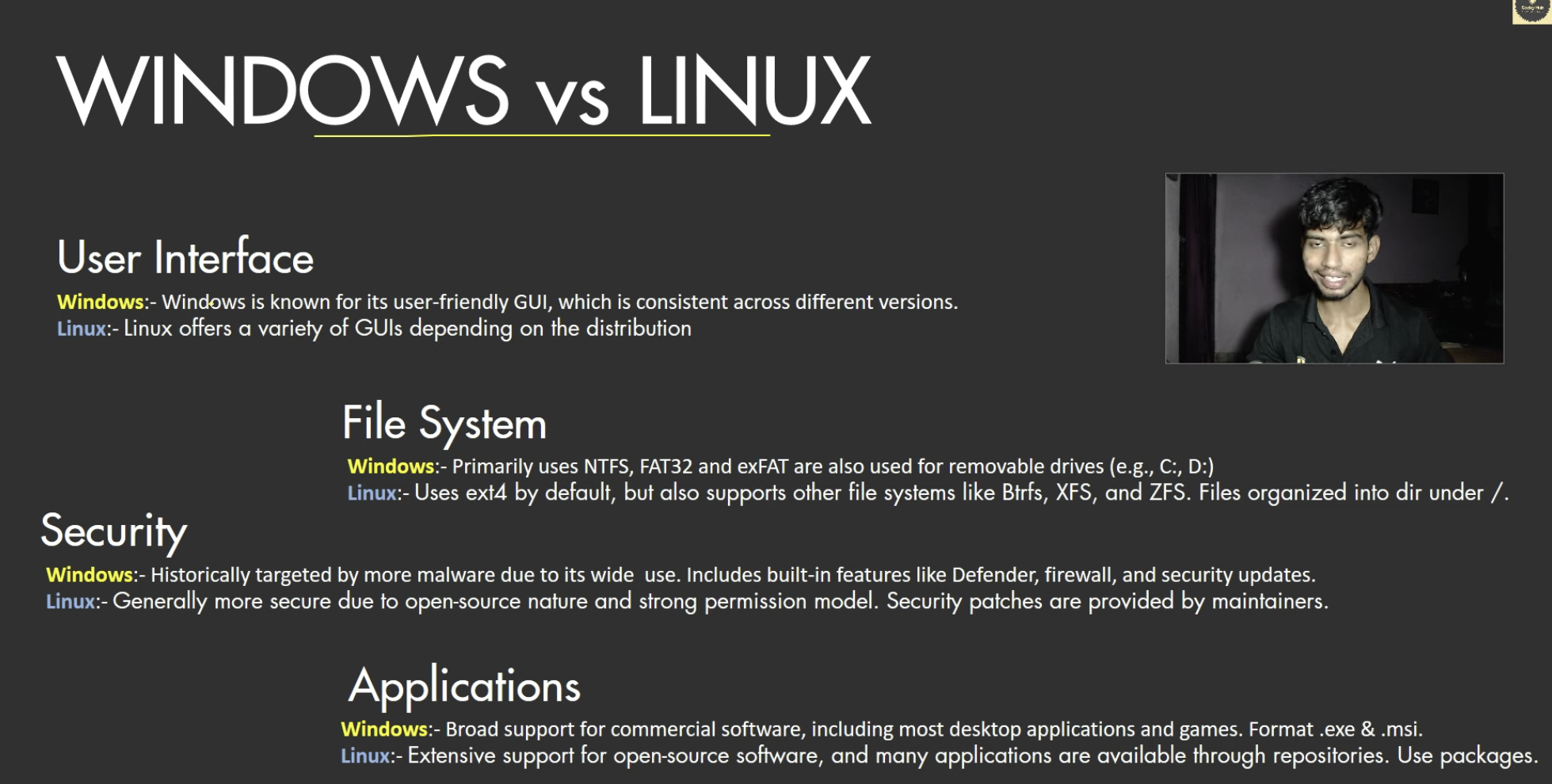
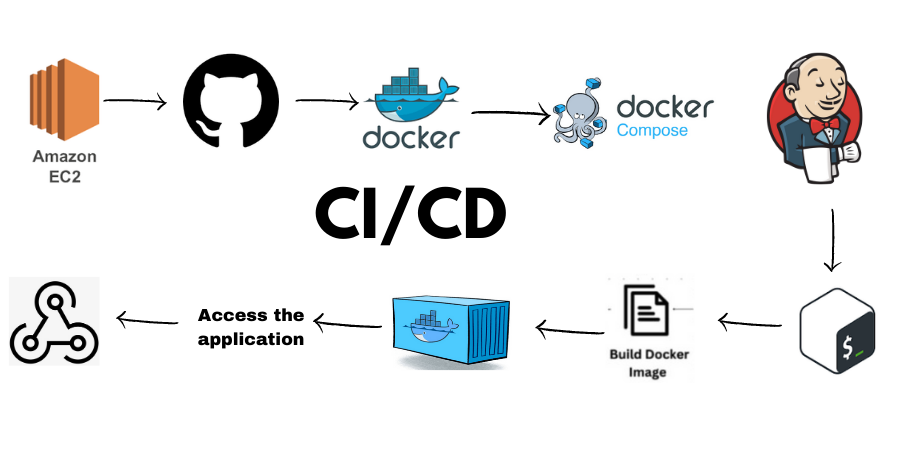
Devops





**CI CD pipeline**



CI/CD pipeline refers to a set of automated processes that help in integrating and delivering code changes continuously and reliably. It consists of two primary concepts: Continuous Integration (CI) and Continuous Deployment/Delivery (CD).

**1. Continuous Integration (CI):**

* CI is the practice of merging code changes from multiple developers into a shared repository frequently, often several times a day.
* Automated tests are run to ensure that the new code doesn't break existing functionality.
* CI tools like Jenkins, GitLab CI, CircleCI, etc., help automate these tasks.

**2. Continuous Delivery/Deployment (CD):**

* **Continuous Delivery**: Once the code passes all tests and builds successfully, it is ready for release into production. However, the deployment is triggered manually.
* **Continuous Deployment**: Code is automatically deployed to production as soon as it passes the tests without any manual intervention.

**Main Steps in a CI/CD Pipeline:**

1. **Source Code Management**: Developers push code changes to a version control system (like Git).
2. **Automated Build**: The pipeline runs an automated build process to compile and package the code.
3. **Automated Testing**: Unit tests, integration tests, and other automated tests are executed to verify code quality.
4. **Deployment**: The code is deployed to staging or production environments.
5. **Monitoring and Feedback**: Monitoring tools ensure that the system is running smoothly, and feedback is collected.

CI/CD pipelines help to:

* Speed up the development process.
* Ensure that code is tested and ready for production frequently.
* Minimize the risk of bugs by detecting issues early.

**SSH**

An **SSH client** is a software application that enables a user to connect to a remote computer or server securely via the **Secure Shell (SSH)** protocol. SSH is widely used for secure access and management of servers, remote systems, and devices over a network, typically the internet.

**Key Features of SSH Client:**

1. **Secure Communication**: SSH encrypts the data exchanged between the client and the server, providing confidentiality and integrity.
2. **Remote Command Execution**: You can execute commands on the remote system as if you were physically present at the machine.
3. **File Transfer**: SSH allows secure file transfers using protocols like **SCP (Secure Copy Protocol)** or **SFTP (Secure File Transfer Protocol)**.
4. **Tunneling and Port Forwarding**: SSH clients can create secure tunnels to forward ports between the client and server, useful for accessing internal services securely.

**Common SSH Clients:**

* **Linux/macOS**: Built-in command-line SSH client (ssh command).
* **Windows**: Tools like **PuTTY**, **Windows PowerShell** (ssh command in newer versions), or **MobaXterm**.
* **Third-Party Tools**: Tools like **Termius** and **Bitvise** provide cross-platform support and a graphical interface.

**How an SSH Client Works:**

1. **Connection Initiation**: The user provides the SSH client with the hostname/IP address and port (default is port 22) of the server they wish to connect to.
2. **Authentication**: The client authenticates with the server using credentials (e.g., username/password) or more secure methods like **SSH keys** (public/private key pairs).
3. **Encrypted Session**: Once authenticated, an encrypted session is established, and the user can start issuing commands, transferring files, or using port forwarding.

**Common Use Cases:**

* **Remote server management**.
* **Cloud infrastructure access** (e.g., accessing AWS, GCP, Azure instances).
* **File transfers** between local and remote systems.
* **Tunneling traffic securely** for internal applications.

Open Jenkins website and follow the below steps

**First install java**

sudo apt update

sudo apt install fontconfig openjdk-17-jre

java -version

openjdk version "17.0.8" 2023-07-18

OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)

OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)

**Install Jenkins**

sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update

sudo apt-get install Jenkins

**You can enable the Jenkins service to start at boot with the command:**

sudo systemctl enable jenkins

**Jenkins will be running in port 8080**

61ba05055526443e802255d3ddf44095 /var/lib/jenkins/secrets/initialAdminPassword

Above is the password stored in specific location