

2-2-1: Examining CI/CD Pipeline Components

After completing this episode, you should be able to:

- Identify and explain common CI/CD pipeline artifacts, given a scenario.

Description: In this episode, the learner will examine various artifacts that are produced during a CI/CD pipeline. We will explore repositories (repos), images, VM images, contain images, packages, and more.

- Describe what a repository (repo) is DevOps
 - o A repository (repo) is a centralized location where code and project files are stored, facilitating version control and collaboration within a CI/CD pipeline.
- Describe the types of repositories
 - o Private repository
 - ♣ A private repository restricts access to specific users or teams, ensuring that proprietary code and sensitive information are kept secure and confidential.
 - o Public Repository
 - ♣ A public repository is accessible to anyone, promoting open collaboration and transparency, but requiring careful management to protect against the exposure of sensitive data and intellectual property.
- Describe what an artifact means in CI/CD pipelines
 - o By-products or outputs that are generated during the software development process.
 - o Each type of artifact has specific roles and applications within the DevOps lifecycle.
- Describe examples of the types of artifacts that can be generated during a CI/CD pipeline
 - o Images
 - ♣ These are static files contain everything needed to run an application (dependencies, libraries, code, environment).
 - ♣ VM (Virtual Machine)
 - ♣ These encapsulate virtualized hardware, operating system, and applications.
 - ♣ Useful for environments were more control of the virtualized hardware and operating system configuration
 - ♣ Containers
 - ♣ Unlike VM images, container use images that contain only the application and its dependencies, but not the OS.
 - ♣ Used to create lightweight, reproducible, and scalable environments that are ideal for microservices architectures
 - o Packages
 - ♣ These are collections of software that are packaged together, allowing for easy distribution and installation.
 - ♣ Examples:
 - ♣ Red Hat Package Manager (RPM)
 - ♣ Used primarily in Red Hat systems, RPM packages contain compiled code, installation scripts, and metadata about the package. In DevOps, RPMs facilitate consistent application deployment across environments.
 - ♣ Debian Packages (deb)
 - ♣ "deb" paackages are similar to RPMs but used in Debian-based systems
 - ♣ These packages help ensure that software installations and upgrades are consistent and traceable across all stages of development and deployment.
 - ♣ ZIP
 - ♣ A compressed archive that can contain any sort of digital files. ZIP files

are often used in DevOps to transfer multiple files as part of software deployments or when passing artifacts between stages in a CI/CD pipeline.

- ♣ tar

- ♣ A UNIX-based utility used for packaging multiple files into a single archive (often tar.gz or tar.bz2 for compressed versions).

- ♣ It's widely used in Linux environments to handle software distributions and backups in DevOps workflows.

- o Flat file

- ♣ This refers to a data file that does not contain any structured relationships (unlike databases).

- ♣ Flat files like configuration files or scripts are used to manage application settings and automate environment setups across different stages of the CI/CD pipeline.