**Components and Workflow Explanation**

1. **Jenkins Pipeline:**
   * **Action:** Initiates the build process by invoking the shared library method setupBuildTool.
   * **Inputs Provided:**
     + toolName (e.g., maven)
     + toolVersion (e.g., 3.8.7)
     + downloadUrl (URL to download the tool from Artifactory)
2. **Shared Library Method (setupBuildTool):**
   * **Purpose:** Manages the verification, downloading, setup, and path configuration of the specified build tool.
   * **Inputs:**
     + toolName: Name of the build tool (e.g., maven, node)
     + toolVersion: Specific version required
     + downloadUrl: Artifactory URL for downloading the tool
3. **Detect Operating System:**
   * **Function:** Determines whether the Jenkins agent is running on Windows or Linux.
   * **Implementation:** Utilize Jenkins environment variables or Groovy's System properties to detect the OS.
   * **Purpose:** Ensures that subsequent steps are executed with OS-specific commands and paths.
4. **Path Setup (Windows vs. Linux):**
   * **Windows:**
     + **Central Tools Directory:** Typically C:\Jenkins\tools\
     + **Workspace Symlink Directory:** C:\Jenkins\workspace\tools\
   * **Linux:**
     + **Central Tools Directory:** /opt/jenkins/tools/
     + **Workspace Symlink Directory:** $WORKSPACE/tools/
5. **Check Central Tools Directory:**
   * **Function:** Verifies if the specified tool and version already exist in the central directory.
   * **Decision Points:**
     + **Yes:** Tool exists
     + **No:** Tool does not exist and needs to be downloaded
6. **If Tool Exists:**
   * **Return Existing Tool Path:**
     + **Windows Example:** C:\Jenkins\tools\maven-3.8.7\bin\mvn.exe
     + **Linux Example:** /opt/jenkins/tools/maven-3.8.7/bin/mvn
   * **Create Symlink in Workspace:**
     + **Windows:** Use mklink to create a directory junction or symbolic link.

batch

Copy code

mklink /J C:\Jenkins\workspace\tools\maven-3.8.7 C:\Jenkins\tools\maven-3.8.7

* + - **Linux:** Use ln -s to create a symbolic link.

bash

Copy code

ln -sfn /opt/jenkins/tools/maven-3.8.7 $WORKSPACE/tools/maven-3.8.7

* + **Update PATH Environment Variable:**
    - **Windows:**

groovy

Copy code

env.PATH = "C:\\Jenkins\\workspace\\tools\\${toolName}-${toolVersion}\\bin;${env.PATH}"

* + - **Linux:**

groovy

Copy code

env.PATH = "${workspaceToolPath}:${env.PATH}"

* + **Output:** Returns the path to the tool's bin directory for use in subsequent build stages.

1. **If Tool Does Not Exist:**
   * **Download Tool from Artifactory:**
     + **Windows:**
       - Use PowerShell or curl to download.

powershell

Copy code

Invoke-WebRequest -Uri $downloadUrl -OutFile "C:\Jenkins\tools\${toolName}-${toolVersion}.zip"

* + - **Linux:**
      * Use curl or wget to download.

bash

Copy code

curl -L -o /opt/jenkins/tools/${toolName}-${toolVersion}.zip ${downloadUrl}

* + **Uncompress Tool Archive:**
    - **Windows:**
      * Use PowerShell's Expand-Archive or a tool like 7-Zip.

powershell

Copy code

Expand-Archive -Path "C:\Jenkins\tools\${toolName}-${toolVersion}.zip" -DestinationPath "C:\Jenkins\tools\${toolName}-${toolVersion}"

* + - **Linux:**
      * Use unzip or tar based on archive type.

bash

Copy code

unzip /opt/jenkins/tools/${toolName}-${toolVersion}.zip -d /opt/jenkins/tools/${toolName}-${toolVersion}

* + **Set Executable Permissions (If Necessary):**
    - **Windows:** Generally not required, but ensure executables are accessible.
    - **Linux:**

bash

Copy code

chmod +x /opt/jenkins/tools/${toolName}-${toolVersion}/bin/\*

* + **Create Symlink in Workspace:** *(Same as in "If Tool Exists" step)*
  + **Update PATH Environment Variable:** *(Same as in "If Tool Exists" step)*
  + **Output:** Returns the path to the tool's bin directory.

1. **Return Build Tool Path to Pipeline:**
   * **Windows Example:** C:\Jenkins\workspace\tools\maven-3.8.7\bin\
   * **Linux Example:** /workspace/tools/maven-3.8.7/bin/

**Detailed Workflow Steps**

1. **Start: Provide Inputs**
   * **Inputs:**
     + toolName: Name of the build tool (e.g., maven, node)
     + toolVersion: Specific version (e.g., 3.8.7, 14.17.0)
     + downloadUrl: Direct URL to download the tool archive from Artifactory
2. **Initialize: Load Shared Library and Define Paths**
   * **Action:** The Jenkins pipeline loads the shared library containing the setupBuildTool method.
   * **Define Paths:** Based on the detected OS, set base directories for tools and workspace symlinks.
3. **Detect Operating System**
   * **Windows:**
     + Paths use backslashes (\)
     + Executable extensions like .exe
   * **Linux:**
     + Paths use forward slashes (/)
     + Executables do not have extensions
4. **Check Central Tools Directory**
   * **Function:** Verify if the tool exists in the central tools directory.
   * **Condition:** Check for the presence of the tool's binary.
5. **Decision Point: Tool Exists?**
   * **Yes:**
     + **Action:** Proceed to set up environment paths and create symlinks.
   * **No:**
     + **Action:** Download and set up the tool as per OS-specific steps.
6. **Download and Uncompress Tool (If Needed)**
   * **Windows:**
     + Use PowerShell commands or Windows-specific utilities to handle downloads and extraction.
   * **Linux:**
     + Utilize shell commands like curl/wget and unzip/tar.
7. **Set Executable Permissions (Linux Only)**
   * **Action:** Ensure that binaries have execute permissions using chmod.
8. **Create Symlink in Workspace**
   * **Purpose:** Provide workspace-specific access to the tool without duplicating the files.
   * **Windows:** Use mklink for directory junctions or symbolic links.
   * **Linux:** Use ln -sfn to create or update symbolic links.
9. **Update PATH Environment Variable**
   * **Purpose:** Make the tool's binaries accessible in the build environment.
   * **Windows:** Prepend the tool's bin directory to env.PATH using Windows path syntax.
   * **Linux:** Prepend the tool's bin directory to env.PATH using Unix path syntax.
10. **Return Build Tool Path to Pipeline**
    * **Output:** The method returns the path to the tool's bin directory, which the Jenkins pipeline can use to execute build commands.