**Automate the Tool Path Setup in Jenkins**

**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)
  + toolbinpath (Path where the tool or executable is present in the downloaded zip)

**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
  + toolbinpath: Location where the tool/executable is present in the extracted files

**3. Detect Operating System:**

* **Function:** Determines whether the Jenkins agent is running on Windows or Linux.
* **Implementation:** Uses Jenkins environment variables or Groovy's System properties to detect the OS.
* **Purpose:** Ensures that subsequent steps use OS-specific commands and paths.

**4. Path Setup (Windows vs. Linux):**

* **Windows:**
  + Central Tools Directory: C:\Jenkins\tools\
* **Linux:**
  + Central Tools Directory: /opt/jenkins/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 at /opt/jenkins/tools/${toolName}-${toolVersion}/${toolBinPath}.
  + **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 symbolic link.
* **Update PATH Environment Variable:**
  + **Windows:** env.PATH = "C:\\Jenkins\\workspace\\tools\\${toolName}-${toolVersion}\\bin;${env.PATH}"
  + **Linux:** env.PATH = "${workspaceToolPath}:${env.PATH}"
* **Output:** Returns the path to the tool's bin directory for use in subsequent build stages.

**7. If Tool Does Not Exist: Retrieve Tool Metadata from Artifactory**

* A request is made to Artifactory to fetch a JSON configuration file that contains metadata about available tools.
* The JSON contains:
  + Compatible OS (e.g., Windows, Linux)
  + Product Name & Version
  + Download SourcePath (URL to fetch the tool)
  + Installer Type (Zip, Tar, etc.)
* **Example JSON Configuration:**

[

{

"compatibleOS": ["windows"],

"Product": {

"Name": "Maven",

"Version": "3.9.5"

},

"SourcePath": "https://artifactory/download",

"InstallerType": "Zip"

}

]

* **Dynamic Selection of Tool URL:**
  + The script parses the JSON file and filters it based on the requested toolName, version, and osType.
  + The correct SourcePath is identified for downloading the tool.

**8. Download and Extract Tool:**

1. **Check if the Tool Already Exists:**
   * Before downloading, check if the tool is already available on the Jenkins agent in a centralized location.
   * If the tool exists, skip downloading.
2. **Determine Centralized Storage Location:**
   * **Linux:** /opt/jenkins/tools/${toolName}/${version}/
   * **Windows:** C:\JenkinsTools\${toolName}\${version}\
3. **Download the Tool:**
   * Fetch the tool from the SourcePath identified in the JSON.
4. **Extract Based on Installer Type:**
   * If InstallerType is "Zip", extract the .zip file.
   * If InstallerType is "Tar", extract the .tar.gz file.

* **Set Executable Permissions (If Necessary):**
  + **Windows:** Ensure executables are accessible.
  + **Linux:** 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.

**9. Identify and Configure Tool Path**

* **Find the Executable File:**
  + **Linux:** Assume the tool's executable is inside the bin/ directory.
  + **Windows:** Search for the .exe file in the extracted folder.
* **Validate Executable Path:**
  + Ensure that the identified tool executable exists after extraction.
* **Set Environment Variable:**
  + Dynamically set the tool path for Jenkins builds.

**10. 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/