**Windows Development Environment Setup Script**

This PowerShell script installs several essential development tools on a Windows machine. It uses Chocolatey, a package manager for Windows, to automate the installation process. It ensures a consistent setup for a development environment by installing commonly used development tools and configuring the necessary environment variables.

**Prerequisites:**

- Windows operating system

- Administrative privileges

**Automated Software Installation Script:**

What This Script Does:

1. Checks if the script is run with administrative privileges.

2. Installs Chocolatey if not already installed.

3. Installs the following software:

- OpenJDK

- Visual Studio Code

- IntelliJ IDEA Community Edition

- Apache Maven

- Node.js

- Postman

- Git

- Docker Desktop

- AWS CLI

- Terraform

4. Sets environment variables for Java and Docker.

5. Verifies the installations by running version commands.

**How to Use:**

1. Open PowerShell as an Administrator.

2. Copy and paste the entire script into the PowerShell window or save it as a `.ps1` file and run it from PowerShell.

3. The script will output the progress and any errors encountered during the installation process.

**Script:**

```powershell

**# Check for Administrative privileges:**

if (!([Security.Principal.WindowsPrincipal][Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltInRole] "Administrator")) {

Write-Host "Please run this script as an Administrator."

exit

}

**# Function to refresh environment variables:**

function Refresh-EnvironmentVariables {

$env:Path = [System.Environment]::GetEnvironmentVariable("Path", [System.EnvironmentVariableTarget]::Machine)

}

**# Function to install a package and check for success:**

function Install-Package {

param (

[string]$packageName

)

Write-Host "Installing $packageName..."

choco install -y $packageName --no-progress

if ($LASTEXITCODE -ne 0) {

Write-Host "Failed to install $packageName. Exiting script."

exit

}

Write-Host "$packageName installed successfully."

Refresh-EnvironmentVariables

}

**# Install Chocolatey:**

Write-Host "Installing Chocolatey..."

Set-ExecutionPolicy Bypass -Scope Process -Force

[System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072

Invoke-Expression ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))

Refresh-EnvironmentVariables

**# Install Java first:**

Install-Package "openjdk"

**# Set JAVA\_HOME environment variable:**

Write-Host "Setting JAVA\_HOME environment variable..."

$javaPath = (Get-Command java).Source.Replace("\bin\java.exe", "")

[Environment]::SetEnvironmentVariable("JAVA\_HOME", $javaPath, [System.EnvironmentVariableTarget]::Machine)

Refresh-EnvironmentVariables

Write-Host "JAVA\_HOME set to $javaPath."

**# Install remaining software one by one:**

Install-Package "visualstudiocode"

Install-Package "intellijidea-community"

Install-Package "maven"

Install-Package "nodejs"

Install-Package "postman"

Install-Package "git"

Install-Package "docker-desktop"

Install-Package "awscli"

Install-Package "terraform"

**# Add Docker to the system path:**

Write-Host "Adding Docker to the system path..."

$dockerPath = "C:\Program Files\Docker\Docker\resources\bin"

[Environment]::SetEnvironmentVariable("Path", $env:Path + ";$dockerPath", [System.EnvironmentVariableTarget]::Machine)

Refresh-EnvironmentVariables

Write-Host "Docker added to the system path."

**# Verify installations:**

Write-Host "Verifying installations..."

$commands = @(

"java -version",

"mvn -version",

"node -v",

"npm -v",

"git --version",

"docker --version",

"aws --version",

"mysql --version",

"terraform -version",

"code --version",

"idea --version"

)

foreach ($command in $commands) {

Write-Host "Running $command..."

try {

Invoke-Expression $command

} catch {

Write-Host "Failed to run $command."

}

}

Write-Host "All installations complete and verified."

**Notes:**

* Ensure you have a stable internet connection during the installation process.
* If any installation fails, the script will stop and output an error message.
* You may need to restart your computer for some environment variable changes to take effect.