***Practical AI: Automating Operations on Amazon Linux with Scripted Intelligence***

### **Explanation of the Script:**

This Python script automates the process of **updating all installed Python packages** using pip. It follows these steps:

1. **Retrieve the list of installed packages:**
   * The script uses subprocess.check\_output() to run the command pip freeze, which lists all installed Python packages and their versions.
   * The output is decoded and split into lines to get individual package names.
2. **Iterate through each package and update it:**
   * The script extracts the package name by splitting each entry at "==" (which separates the package name from its version).
   * It prints a message indicating that the package is being updated.
3. **Upgrade the package:**
   * The script runs pip install --upgrade <package\_name> using subprocess.check\_call().
   * If the update is successful, it prints a confirmation message.
   * If an error occurs during the update, it catches the exception and prints an error message.
4. **Execute the function:**
   * The script includes a standard Python if \_\_name\_\_ == "\_\_main\_\_" condition to ensure that update\_packages() runs only if the script is executed directly.

import subprocess

import sys

def update\_packages():

# List installed packages using pip

installed\_packages = subprocess.check\_output([sys.executable, "-m", "pip", "freeze"]).decode().splitlines()

# Iterate through each package and update it

for package in installed\_packages:

package\_name = package.split("==")[0]

print(f"Updating {package\_name}...")

try:

# Upgrade the package to the latest version

subprocess.check\_call([sys.executable, "-m", "pip", "install", "--upgrade", package\_name])

print(f"{package\_name} has been successfully updated.")

except subprocess.CalledProcessError as e:

print(f"Error updating {package\_name}: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

update\_packages()