utils/helpers.py

import shutil

import subprocess

import logging

logger = logging.getLogger(\_\_name\_\_)

def check\_command\_exists(command):

exists = shutil.which(command) is not None

logger.debug(f"Checking if command '{command}' exists: {'Found' if exists else 'Not Found'}")

return exists

def run\_subprocess(command, cwd=None, capture\_output=True, text=True):

logger.debug(f"Running subprocess: {' '.join(command)} | CWD: {cwd}")

try:

result = subprocess.run(

command,

cwd=cwd,

capture\_output=capture\_output,

text=text,

check=True

)

if capture\_output:

logger.debug(f"Subprocess stdout: {result.stdout}")

logger.debug(f"Subprocess stderr: {result.stderr}")

return result.stdout if capture\_output else None

except subprocess.CalledProcessError as e:

logger.error(f"Error running command: {' '.join(command)}")

logger.error(f"Return Code: {e.returncode}")

if capture\_output:

logger.error(f"Stdout: {e.stdout}")

logger.error(f"Stderr: {e.stderr}")

```

I made the following changes:

- Added a check for `capture\_output` before attempting to log the stdout and stderr of the subprocess.

- Added a check for `capture\_output` before returning the stdout of the subprocess. If `capture\_output` is `False`, the function returns `None`.