Windows DevOps Coding Challenge

Automated Install, Build, Run, and Smoke Test on Windows with PowerShell, Pester, and CI

Difficulty	Intermediate
Timebox	~ 180 minutes
Platforms	Windows 10/11

1) Overview

You will create minimal web application (e.g., .NET) exposing a /health endpoint which returns HTTP code 200, and writes a log event when the endpoint is hit (GET).

You will create simple unit test(s) for your application (e.g. using NUnit and/or Pester).

Your further tasks are to automate environment setup, build the application, run it on Windows, validate with smoke tests. You can use CI/CD (e.g. GitHub Actions) or run only locally. Use PowerShell for scripting language. For running on GitHub, a free account with a public repository can be used.

NOTE: Running on GitHub will be beneficial for the final evaluation.

Objectives for the automation part

1. Bootstrap Windows environment (e.g. using winget, if running on GitHub)
2. Build the app (run unit tests if added)
3. Run the app on a configurable port as a background process or a service, on Windows
4. Write Pester smoke tests (port open, /health returns 200, logs are created)
5. Preferably a CI/CD on GitHub Actions, using windows-latest runner

2) Suggested Code Structure (example)

```
windows-devops-challenge/
■■ app/
■   ■■ src/
■   ■   ■■ Program.cs                  # minimal web application exposing /health endpoint
■   ■   test/
■   ■   ■■ProgramTests.cs              # unit tests
■   ■■ README.md
■■ scripts/
■   ■■ 00-bootstrap.ps1                # tool installation, version verification (needed on GitHub)
■   ■■ 01-build.ps1                    # build and unit test
■   ■■ 02-run.ps1                      # run background (or service)
■   ■■ 03-test.ps1                     # run Pester smoke tests and create test report
■   ■■ 99-cleanup.ps1                  # stop process/service, clean artifacts (optional) ■■ tests/
■   ■■ Smoke.Tests.ps1                 # Pester tests
■   ■■ TestConfig.psd1                 # port, service name, etc. (optional)
■■ .github/
■   ■■ test-workflow.yml               # GitHub Actions workflow (Windows)
■■ logs/                               # runtime logs + test results (artifact)
■■ .gitignore
■■ README.md                           # Description of the solution
```

3) Additional Instructions

- Preferably use Windows 10/11 or Windows
- Preferably use PowerShell 7+ (pwsh)
- The suggested code structure is just an example, which is not working out of the box, the actual structure will depend on the development tools used.
- Fill free to use any online sources, documentation, and AI (Copilot, ChatGPT, etc.)

4) Evaluation

1. Completeness – All core steps automated locally and in CI

2. Clarity – Readable scripts, parameters documented, clear error messages/logs

3. Testing – Pester covers port, health, and logs; results exported (e.g. NUnit XML)

4. Automation – installation of additional tools and cleanup, idempotence, retries where needed

5. Reasoning – Short write-up on assumptions, trade-offs, and next improvements

5) Submission Instructions

• Provide a link to your GitHub repository or a zip archive of the solution

• Include logs/ and pester.xml from a local run (or CI artifacts link)

• Add a short README section describing decisions and any known issues.