# One application through GitHub Actions

DevOps & infrastructure | Intermediate

Session M4-01 | 3 hours

# Agenda

| Time | Topic |
| --- | --- |
| 0:00 to 0:08 | Baseline and Actions model |
| 0:08 to 0:18 | Generate CI |
| 0:18 to 0:28 | Design deployment |
| 0:28 to 0:40 | Repair workflow failures |
| 0:40 to 0:52 | Validate a JavaScript action |
| 0:52 to 1:00 | Review and lab handoff |

# Start with repository facts

`copilot-webapp` already defines the contract:

```
npm ci
npm test
npm run lint
npm run build
test -f dist/app.js
```

The workflow should call these commands. It should not invent another build, artifact path, or runtime.

# A workflow is code with an execution boundary

| Review | Question |
| --- | --- |
| Trigger | Who can cause this code to run? |
| Runner | Which tools and trust boundary apply? |
| Permissions | What can the token read or change? |
| Commands | Do they exist in the repository? |
| Artifacts | Which files leave the job? |
| Environment | Which approval gate applies? |

# Generate CI from the application

```
Read package.json, tests, and scripts/build.js. Create ci.yml for pull requests
and pushes to main. Use Node.js 20, npm ci, existing lint, test, and build scripts,
npm caching, contents: read, and a seven-day dist artifact.
```

Then check both YAML syntax and application behavior:

```
ruby -e 'require "yaml"; Psych.parse_stream(File.read(ARGV[0]))' ci.yml
npm test && npm run lint && npm run build
```

# Deployment stays a design exercise

The same application moves through:

```
build → staging → production
```

The workflow references environments and secrets. Production reviewers are configured in repository settings. Placeholder commands must not imply that a deployment ran.

# Syntax and behavior fail differently

`broken-ci.yml` does not parse.

`broken-deploy.yml` parses but contains a stale runner, a bad Git ref, a wrong secret name, and no checkout.

Ask for the failed-step output before the repair.

# Tie each repair to a failure

```
Observation: the parser fails near Setup Node.js.
Change: align the step indentation.
Proof: parse the file again.
```

For behavior faults, cite the deployment specification or repository facts, then check the repair.

# Validate both action outcomes

```
env INPUT_THRESHOLD=0 ... node src/main.js
env INPUT_THRESHOLD=101 ... node src/main.js
```

The first run must pass and write a summary. The second must exit with code 1.

One successful run does not show that the action enforces the threshold.

# Access and checks

GitHub Copilot access is required. Repository access and Actions are optional.

For each case, inspect:

- command or review;
- result;
- changed files;
- the next action.

# Lab

Create CI, design deployment, repair failures, and validate the action against `copilot-webapp`.