GitHub Actions: From Zero To Hero

Slides up at scottsauber.com

scottsauber

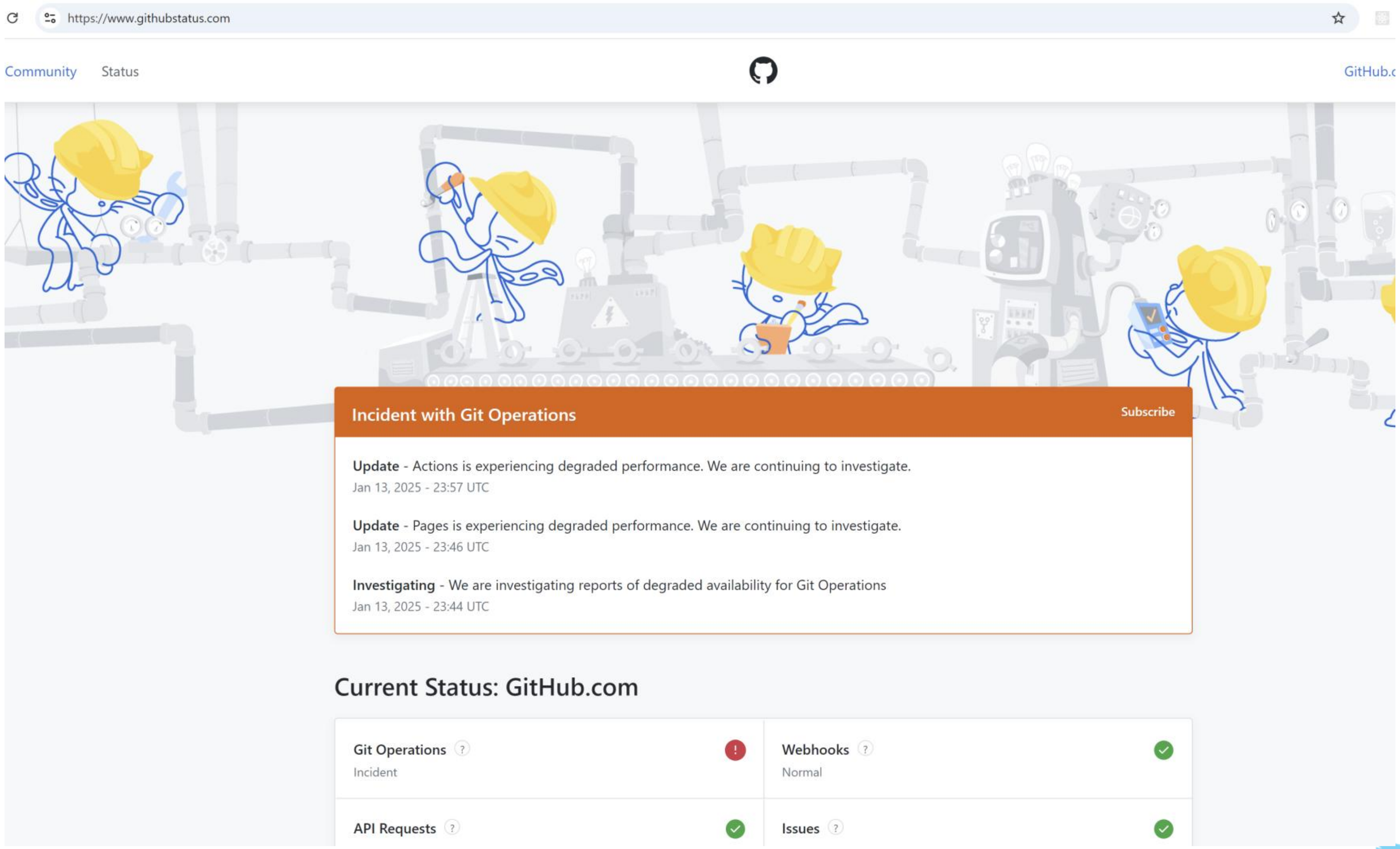
What you need

- Git
- GitHub account
- An Editor (VS Code, Visual Studio, JetBrains IDE, etc)
- Search for a GitHub Actions extension in your editor of choice
- Fork this repo https://github.com/scottsauber/github-actions-workshop
- Optional: .NET 9 (if you want to run the app locally, but not needed)

What we all need

- GitHub to not go down
- GitHub Actions to not go down
- The conference internet to not go down
- 🙏 🙏 🙏
- (I do have recordings but that’s less fun)

A few months ago ~4 hours before this workshop…


https://www.githubstatus.com

Community Status GitHub.c

Incident with Git Operations

Subscribe

Update - Actions is experiencing degraded performance. We are continuing to investigate.
Jan 13, 2025 - 23:57 UTC

Update - Pages is experiencing degraded performance. We are continuing to investigate.
Jan 13, 2025 - 23:46 UTC

Investigating - We are investigating reports of degraded availability for Git Operations
Jan 13, 2025 - 23:44 UTC

Current Status: GitHub.com

Git Operations ? Incident !	Webhooks ? Normal ✓
API Requests ? ✓	Issues ? ✓

scottsauber

Audience

- Anyone interested in GitHub
- People interested in DevOps but rarely/never get to do it
- Already know Git
- If you have questions, ask! Otherwise this is gonna be a boring 4 hrs

Poll

- How many people using GitHub already?
- How many are using GitHub Actions?
- How many feel like they're pretty intermediate to advanced wit GHA?
- What other CI/CD tools are people using?
- Why are you here? What do you want to learn?

Agenda

- What is CI and the two CDs?
- What are GitHub Actions?
- GitHub Actions concepts
- Configuring Optimal GitHub settings
- Creating PR Verify workflow
- Creating CI workflow
- Cron Jobs
- Variables and Secrets
- Reusing workflows
- Things every CI/CD workflow should have

Goals

- Understand what GitHub Actions are
- Get experience using GitHub Actions
- Few takeaways for experienced GitHub Actions users

Who am I?

- Director of Engineering at Lean TECHniques
- Microsoft MVP
- Dometrain Author
- Redgate Community Ambassador
- Co-organizer of Iowa .NET User Group

CI/CD Pipelines

What is Continuous Integration?

- Automated verification of your application
- Generates artifacts
- Compiles the app
- Runs the tests
- Independent witness – eliminates “works on my machine”

What is Continuous Delivery?

- Takes artifacts from CI and deploys them automatically
- Doesn’t deploy all the way to Production
- Deploying to Production is a button click

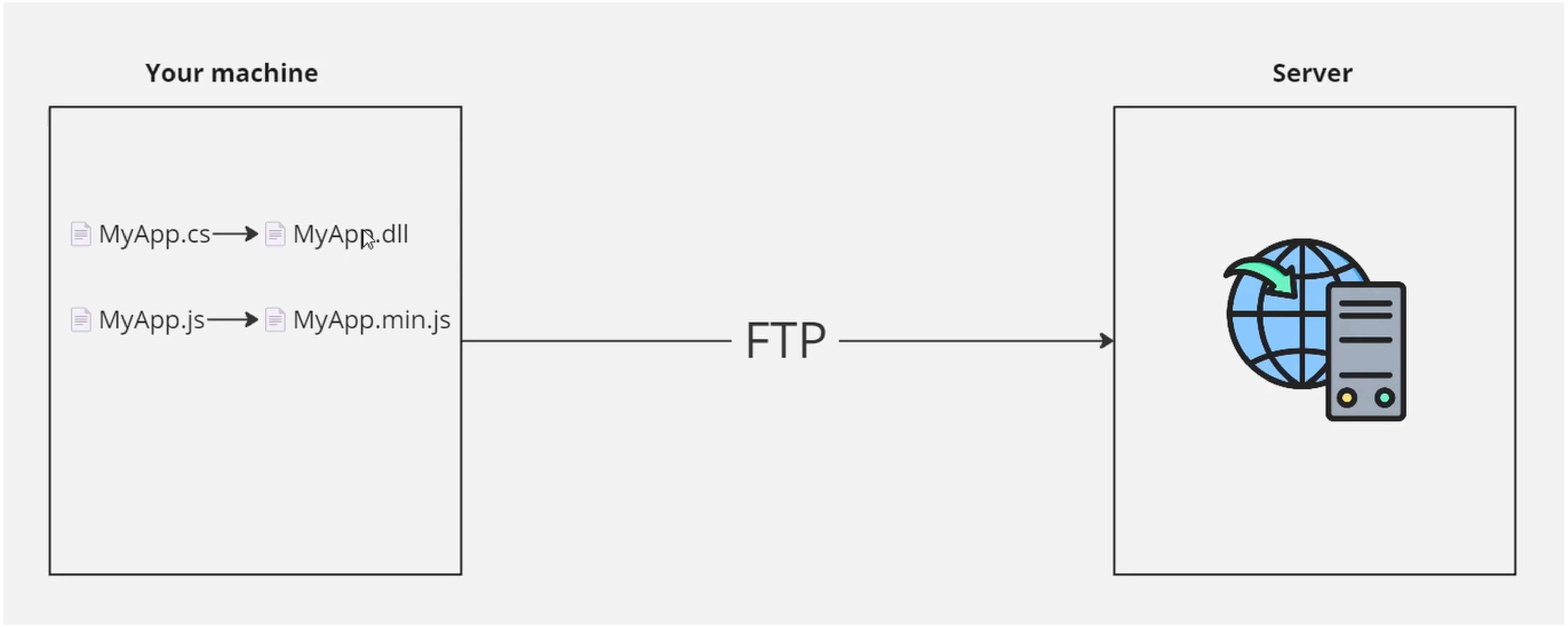
What is Continuous Deployment?

- Deploys all the way to Production automatically
- If the pipeline is green, it’s going to Production

Why CI/CD?

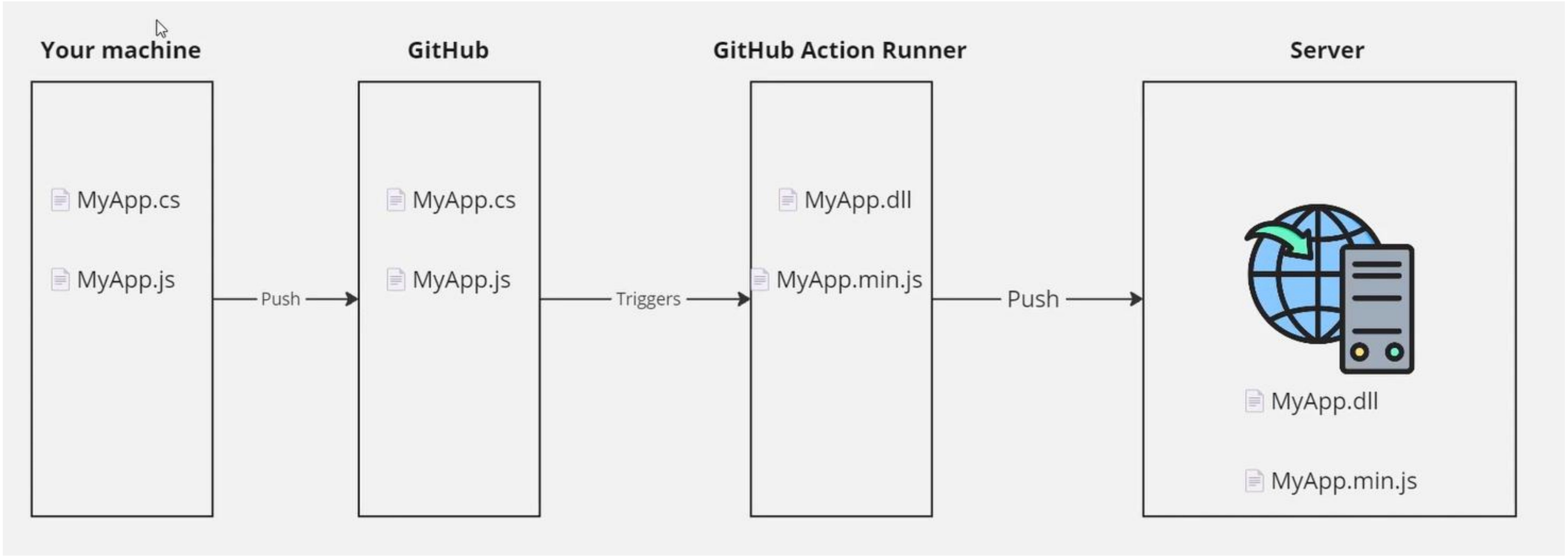
- Avoid manual steps (chances for mistakes)
- Repeatable
- Auditable
- Humans need less permissions

Before CI/CD



Your machine
MyApp.cs
MyApp.dll
MyApp.js
MyApp.min.js
FTP
Server

After CI/CD


Your machine
MyApp.cs
MyApp.js
Push
GitHub
MyApp.cs
MyApp.js
Triggers
GitHub Action Runner
MyApp.dll
MyApp.min.js
Push
Server
MyApp.dll
MyApp.min.js

Confident Green

- If our build passes – why aren’t we shipping to Production?
- Likely lack of confidence or automation
- Likely missing automated tests or zero downtime deployments
- Let’s fix that
- Ok now why?
- Repeat

Ideal CI Pipeline?

- Restore Packages
- Compile
- Test
- Format
- Linting
- Security Scans
- Upload Artifacts
- Alerting on Failure

Ideal CD Pipeline?

- Download Artifacts
- Deploy Artifacts (IAC, DB changes, Application)
- Zero Downtime Deployments
- Smoke Tests
- Security Scans
- Alerting on Failure

CD Today?

- Not going to cover CD today
- Complexity with deploying to Azure
- Workshop link for a workshop I’ve done on “Deploying a .NET 9 app to Azure using GitHub Actions and Bicep” - https://github.com/scottsauber/workshop-dotnet-azure-github-bicep
  - This is an all day workshop
- Also you can buy my Dometrain course – my kids’ basketball traveling teams thank you

Questions about CI/CD?

GitHub Actions

What is GitHub?

- Most popular place for storing source code
- Both public and private
- 80% of our clients are on GitHub, ~20% on Azure DevOps
  - Some moved from BitBucket in last 2-3 years

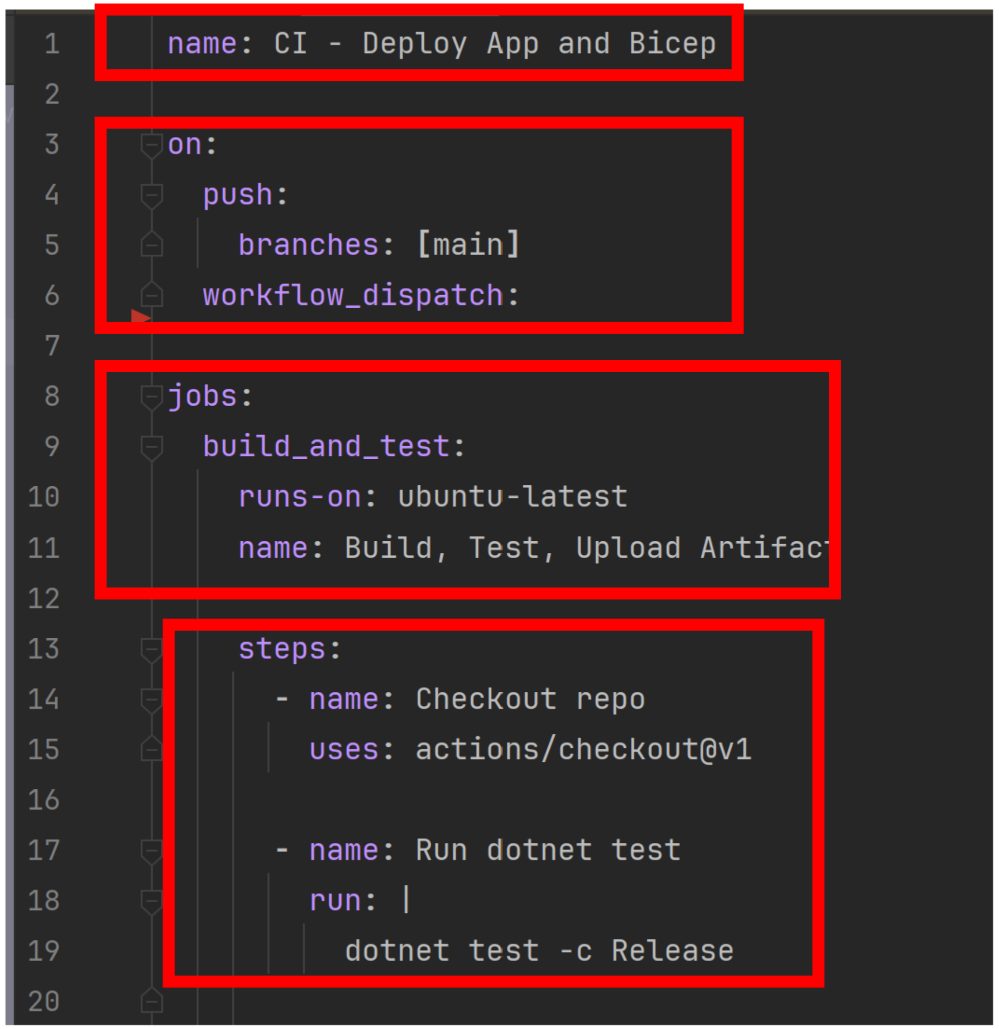
What are GitHub Actions?

- Built into GitHub
- Thing Doer on a trigger
- Trigger could be PR, push to a branch, open an issue, cron, etc
- Usually used to automatically build and deploys your application
- ~70% of our clients are using GitHub Actions
  - Most of these have moved in last 2-3 years (Aug 2018 GHA came out)

GitHub Actions Concepts

- Steps – individual actions to be executed (ie restore packages, compile, etc)
- Jobs – a series of Steps
- Workflows – a series of Jobs
- Triggers – something that kicks off the workflow
- Inputs – parameters to customize a job
- Secrets – sensitive data store in GitHub, can be leveraged in a Workflow
- Runners – Virtual Machines that run Jobs, could be GH-hosted or self-hosted

Example



```yaml
name: CI - Deploy App and Bicep

on:
  push:
    branches: [main]
  workflow_dispatch:

jobs:
  build_and_test:
    runs-on: ubuntu-latest
    name: Build, Test, Upload Artifac

    steps:
      - name: Checkout repo
        uses: actions/checkout@v1

      - name: Run dotnet test
        run: |
          dotnet test -c Release

```

scottsauber

Questions about GitHub Actions?

Live Demo

10 minute break
Then Module 3

Questions about Module 3?

Optimal* GitHub Settings

* synonym for “my opinions”

Optimal GitHub settings

- Repo => Settings
- Pick 1 merge strategy – I use Squash bc most people suck at making a good history
- ✅ Always suggest updating pull request branches
- ✅ Allow auto-merge
- ✅ Automatically delete head branches (GitHub flow or TBD)
- ✅ Configure required status checks as Ruleset (note: merge first)

Live Demo

Questions about Optimal GitHub Settings?

Module 5:
Let’s merge the PR
and set up a required rule set

Questions about Required Rule Sets?

Module 6: CI Workflow On Your Own

Module 6: Review

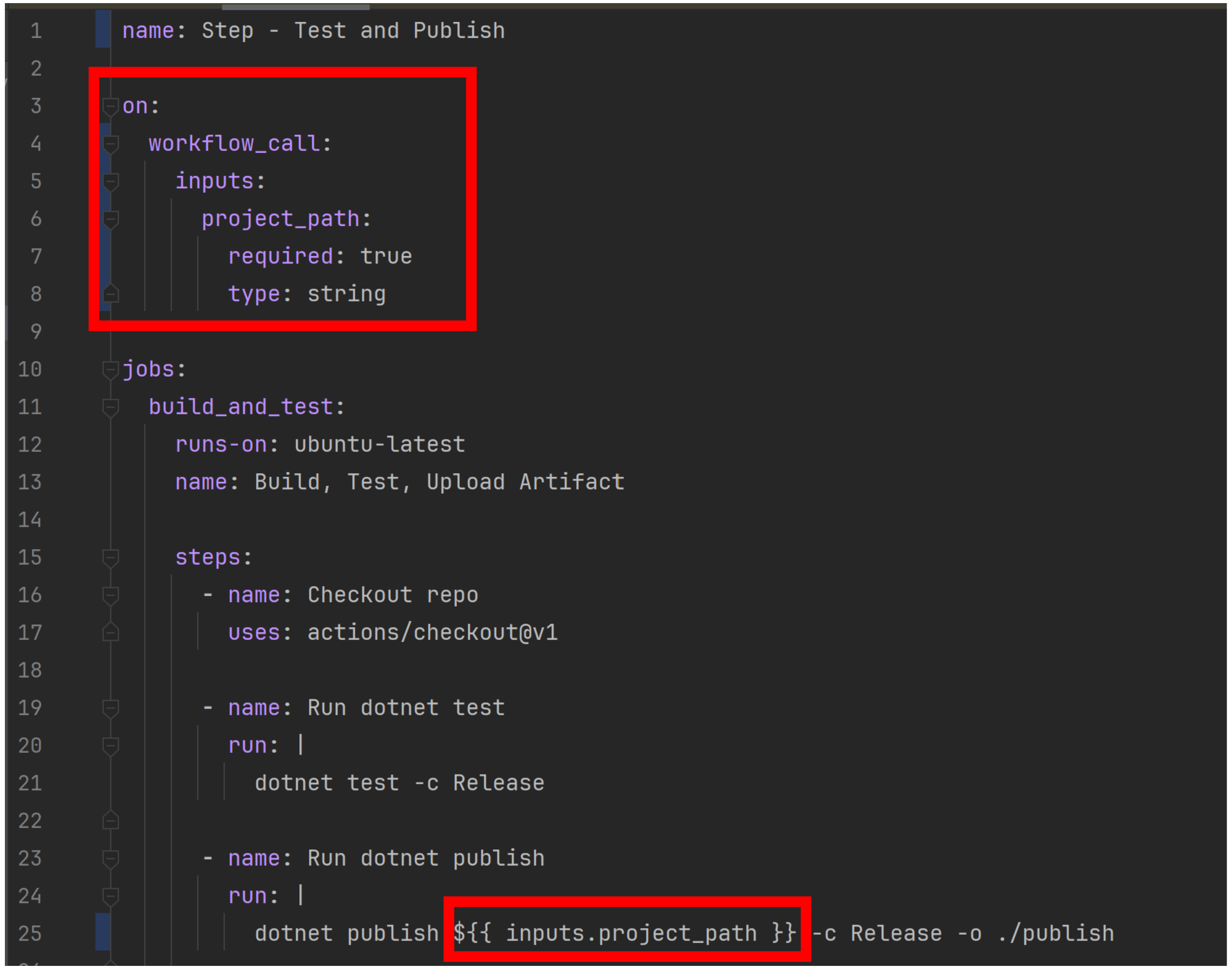
5 Minute Break?

Reusable Workflows

Reusing Workflows

- Copying Pasting YAML feels kinda bad
- GitHub Actions allows reusing workflows via `workflow_call` trigger

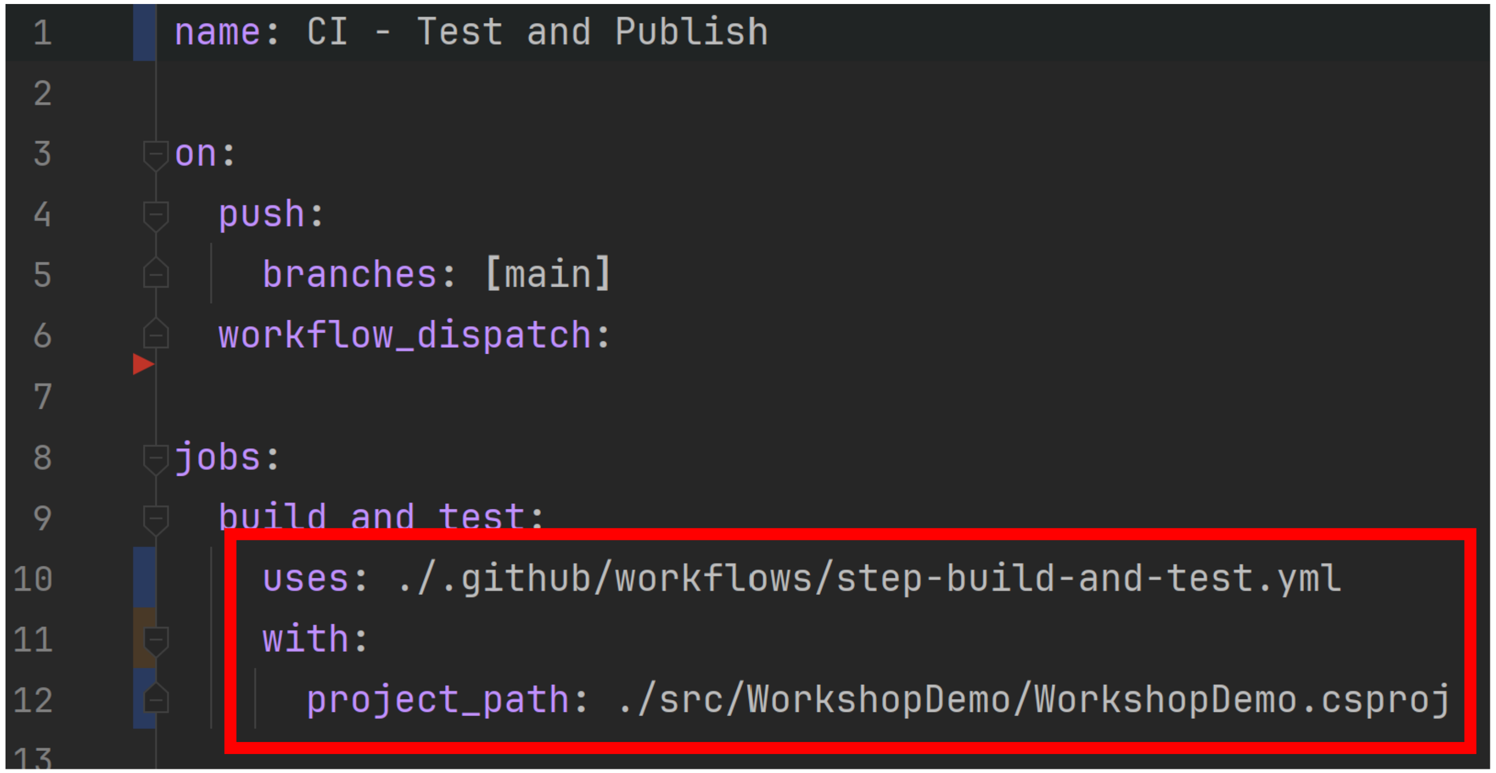
How do I reuse workflows?



```yaml
name: Step - Test and Publish

on:
  workflow_call:
    inputs:
      project_path:
        required: true
        type: string

jobs:
  build_and_test:
    runs-on: ubuntu-latest
    name: Build, Test, Upload Artifact

    steps:
      - name: Checkout repo
        uses: actions/checkout@v1

      - name: Run dotnet test
        run: |
          dotnet test -c Release

      - name: Run dotnet publish
        run: |
          dotnet publish ${{ inputs.project_path }} -c Release -o ./publish
```

scottsauber

Consume reusable workflow

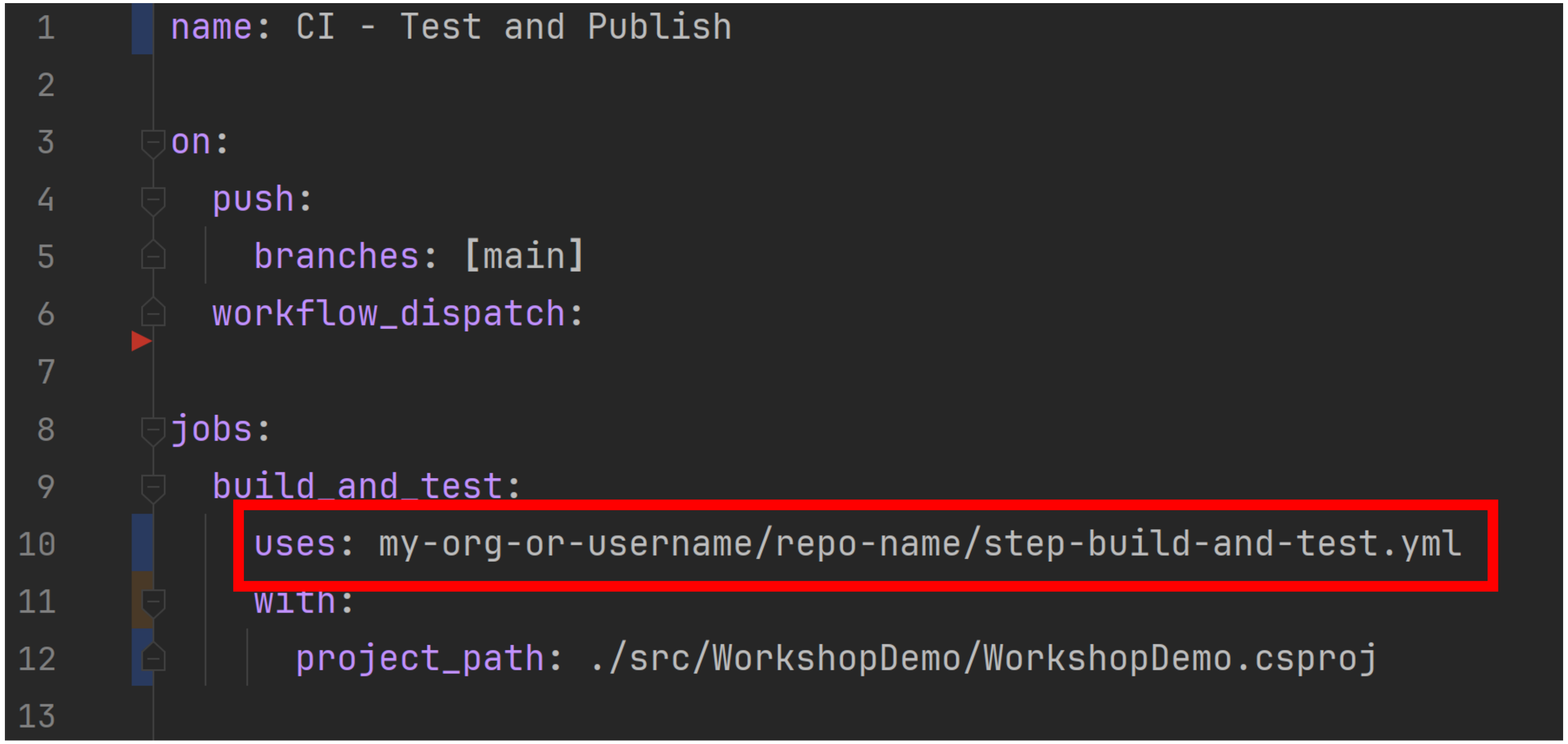


```
name: CI - Test and Publish

on:
  push:
    branches: [main]
  workflow_dispatch:

jobs:
  build_and_test:
    uses: ./.github/workflows/step-build-and-test.yml
    with:
      project_path: ./src/WorkshopDemo/WorkshopDemo.csproj
```

scottsauber

Module 8: Reusable Workflows

Consume reusable workflow from another repo



```yaml
name: CI - Test and Publish

on:
  push:
    branches: [main]
  workflow_dispatch:

jobs:
  build_and_test:
    uses: my-org-or-username/repo-name/step-build-and-test.yml
    with:
      project_path: ./src/WorkshopDemo/WorkshopDemo.csproj

```

scottsauber

Module 10:
Reusable Workflows in another repository

Questions about reusable workflows?

Module 10:
Reusable Workflows in another repository

Secrets

Secrets

- Similar to Inputs, allows you to pass in dynamic values
- Secrets are wildcarded out of the build logs
- They are write only, not read
- Note: you could still exfiltrate secrets via API calls, text files, etc
- Could be Client IDs and Secrets, API Keys, Connection Strings, etc

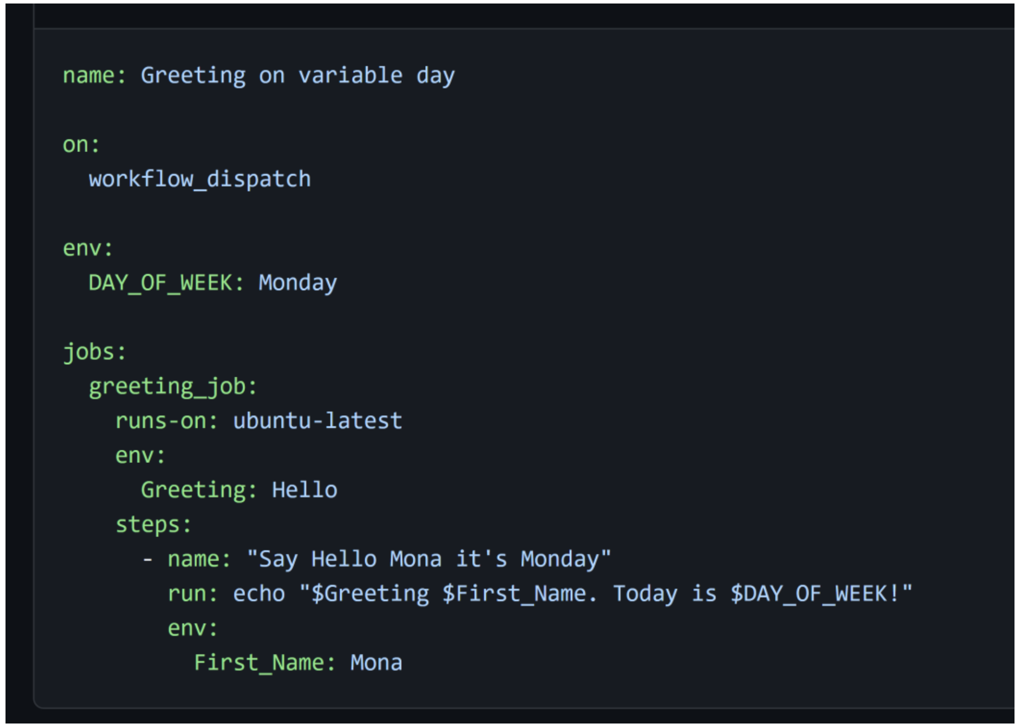
Module 12: Secrets

Questions about Secrets?

Variables

Variables

- For use inside the same workflow file
- Store common paths, versions, or any common string used throughout the workflow



```yaml
name: Greeting on variable day

on:
  workflow_dispatch

env:
  DAY_OF_WEEK: Monday

jobs:
  greeting_job:
    runs-on: ubuntu-latest
    env:
      Greeting: Hello
    steps:
      - name: "Say Hello Mona it's Monday"
        run: echo "$Greeting $First_Name. Today is $DAY_OF_WEEK!"
        env:
          First_Name: Mona
```

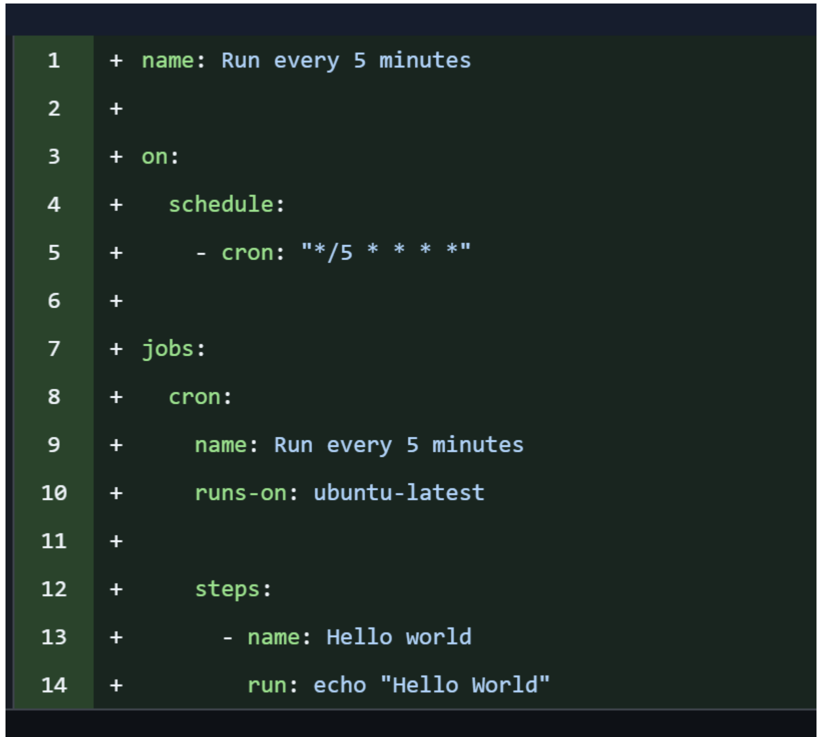
scottsauber

Questions about Variables?

Random GHA Tips

Cron Jobs

- Not meant to be an enterprise scheduler
- No guarantees it runs the time you tell it to
- I’ve seen it run consistently, but up to 15 minutes later



```
+ name: Run every 5 minutes
+
+ on:
+   schedule:
+     - cron: "*/5 * * * *"
+
+ jobs:
+   cron:
+     name: Run every 5 minutes
+     runs-on: ubuntu-latest
+
+     steps:
+       - name: Hello world
+         run: echo "Hello World"
```

scottsauber

Cron Jobs

- Useful for running security scans for repos that don’t get touched very often
- But also run security scans on each change
- Don’t use this to run a daily build, run a build on every commit

Environments

- Allow you to define the environments for deploying your application
- Useful to see what’s deployed successfully
- Allows you to set “Required Approvers” for things like the Production environment
- Allows you to use environment secrets

Live Demo of Environments

GitHub Actions Hero

- https://github-actions-hero.vercel.app/

Pinning Dependencies

- Using ubuntu-latest will break your app
- Instead – pin to things like ubuntu-22.04
- Likewise can pin dependencies
- - uses: actions/checkout@v4.2.2 …. becomes
  - uses: actions/checkout@11bd71901bbe5b1630ceea73d27597364c9af683
- Prevents a malicious actor who gains control of repo (supply chain attack) and changes what 4.2.2 means
- But also promotes consistent builds
- But comes at cost of no free upgrades with a @v4

scottsauber

Breaking Changes

- The downside of not being on the latest – you might not have the latest security fixes
- https://github.com/actions/upload-artifact/issues/602

Bonus: Git Aliases

- https://github.com/scottsauber/dotfiles/blob/main/.bashrc
- gnewbr
- gcpr
- gas
- gcp
- pr
- gpo
- gcopm

More Random tips

- Leverage ChatGPT/Copilot/Cursor/Claude Code – they're really good at YAML
- path filters
- Sparse checkouts

More Random tips

- When deploying to the cloud – use Federated Credentials (Azure) or Federated Identity (AWS)
- Passwordless
- Allows you to authenticate and say “this org and repo can deploy to this account”
- This is something we check on our Azure Cloud Health Check and 90% of companies aren’t doing this

Resources

- This slide deck
- https://github.com/scottsauber/github-actions-workshop
- https://github.com/scottsauber/workshop-dotnet-azure-github-bicep

Schedule time with me:

Questions?

ssauber@leantechniques.com
@scottsauber.com on Bluesky
@scottsauber on Twitter

Thanks!

Slides up at scottsauber.com

scottsauber