

GitHub Actions: From Zero To Hero

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

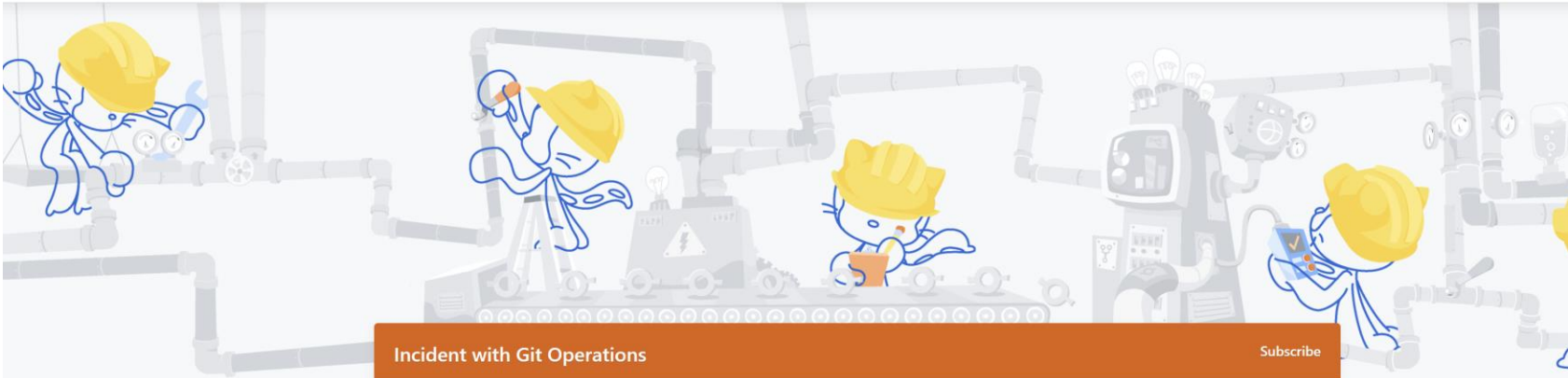
- GitHub to not go down
- GitHub Actions to not go down
- The conference internet to not go down
- 🙏 🙏 🙏
- (I do have recordings but that's less fun)

Last night at 7pm...

https://www.githubstatus.com

Community Status

GitHub.com



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 ?	✓

Audience

- Anyone interesting in GitHub
- People interest 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

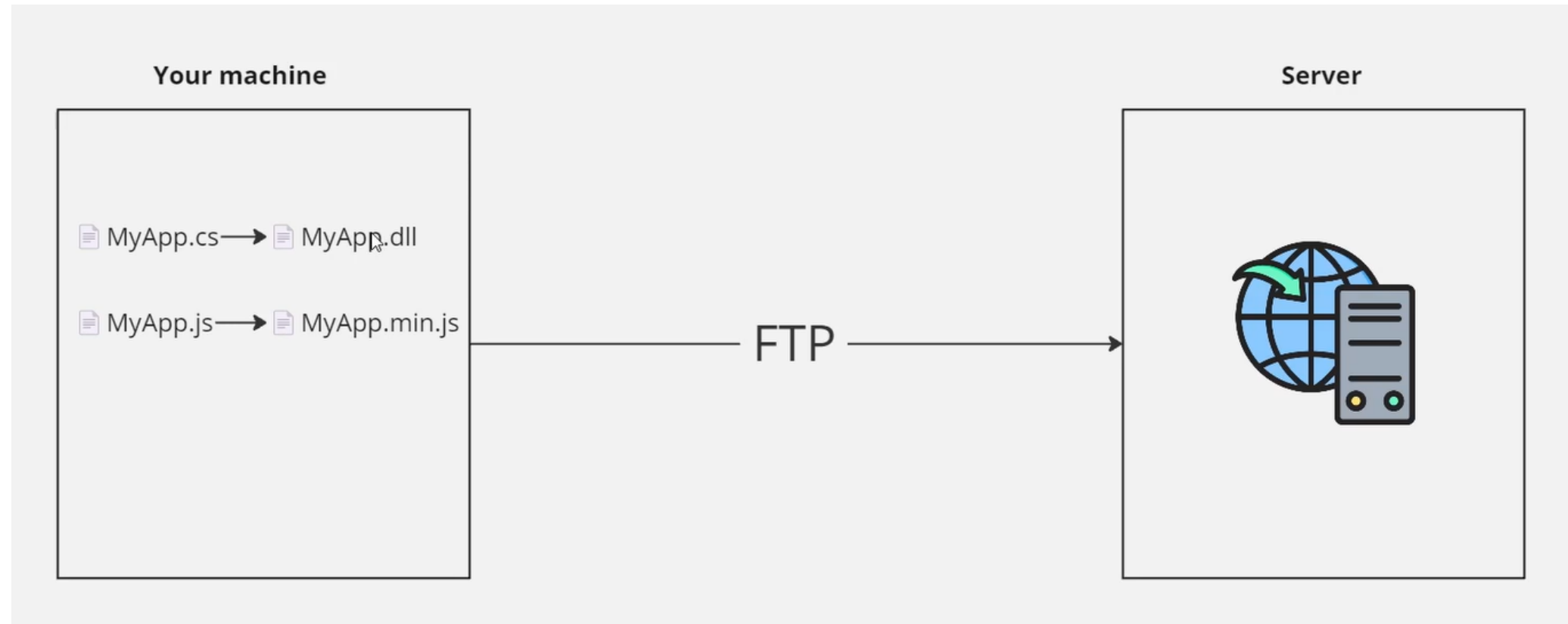
What is Continuous Deployment?

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

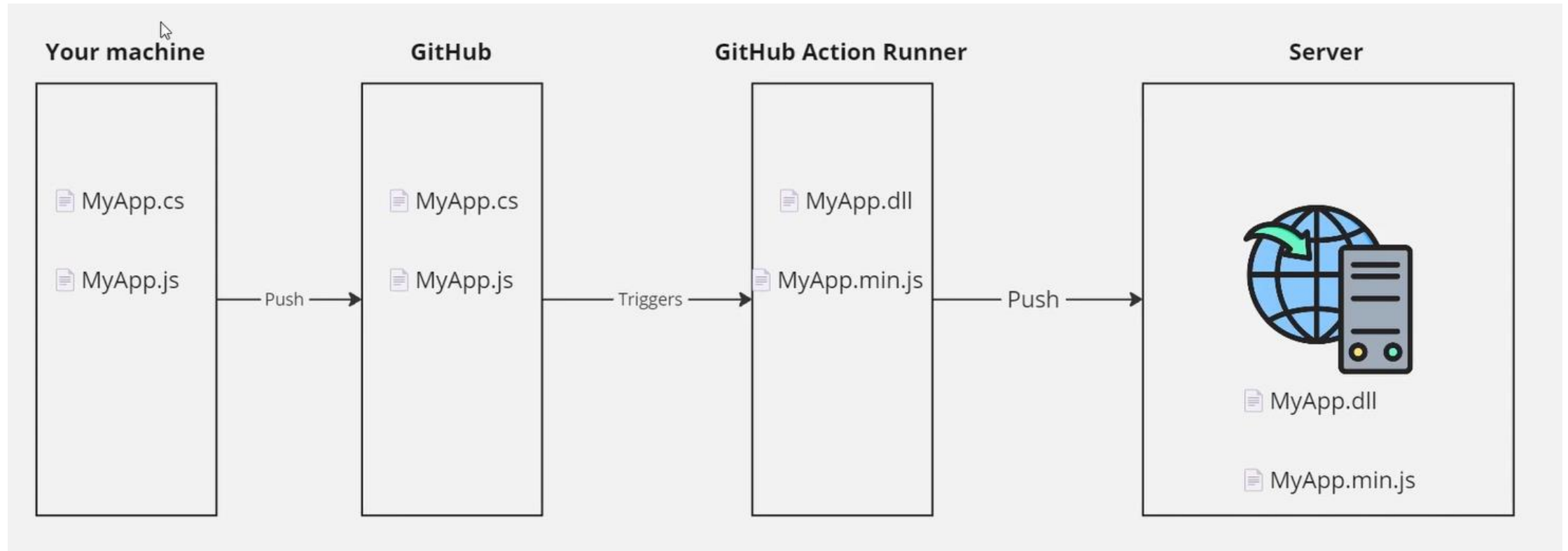
Why CI/CD?

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

Before CI/CD



After CI/CD



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 This Afternoon

- Not going to cover CD today
- Complexity with deploying to Azure
- Advanced GitHub Actions workshop this afternoon
- 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

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

```
1  name: CI - Deploy App and Bicep
2
3  on:
4    push:
5      branches: [main]
6    workflow_dispatch:
7
8  jobs:
9    build_and_test:
10     runs-on: ubuntu-latest
11     name: Build, Test, Upload Artifact
12
13     steps:
14       - name: Checkout repo
15         uses: actions/checkout@v1
16
17       - name: Run dotnet test
18         run: |
19           dotnet test -c Release
20
```

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?

```
1 name: Step - Test and Publish
2
3 on:
4   workflow_call:
5     inputs:
6       project_path:
7         required: true
8         type: string
9
10  jobs:
11    build_and_test:
12      runs-on: ubuntu-latest
13      name: Build, Test, Upload Artifact
14
15      steps:
16        - name: Checkout repo
17          uses: actions/checkout@v1
18
19        - name: Run dotnet test
20          run: |
21            dotnet test -c Release
22
23        - name: Run dotnet publish
24          run: |
25            dotnet publish ${ inputs.project_path } -c Release -o ./publish
```

Consume reusable workflow

```
1  name: CI - Test and Publish
2
3  on:
4    push:
5      branches: [main]
6    workflow_dispatch:
7
8  jobs:
9    build and test:
10      uses: ../.github/workflows/step-build-and-test.yml
11      with:
12        project_path: ./src/WorkshopDemo/WorkshopDemo.csproj
13
```

Module 8: Reusable Workflows

Consume reusable workflow from another repo

```
1  name: CI - Test and Publish
2
3  on:
4    push:
5      branches: [main]
6    workflow_dispatch:
7
8  jobs:
9    build_and_test:
10      uses: my-org-or-username/repo-name/step-build-and-test.yml
11      with:
12        project_path: ./src/WorkshopDemo/WorkshopDemo.csproj
13
```

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

```
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
```

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

```
1 + name: Run every 5 minutes
2 +
3 + on:
4 +   schedule:
5 +     - cron: "*/5 * * * *"
6 +
7 + jobs:
8 +   cron:
9 +     name: Run every 5 minutes
10 +    runs-on: ubuntu-latest
11 +
12 +    steps:
13 +      - name: Hello world
14 +        run: echo "Hello World"
```

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

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 – o1 model (advanced reasoning) when asking for the 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>

Questions?

Follow up: ssauber@leantechniques.com

Thanks!