SESSION

GitHub Actions: From Zero to Hero

Scott Sauber
Director of Engineering, Lean TECHniques Inc

DEVintersection Conference

next GenAI CONFERENCE

Cybersecurity INTERSECTION

Join us on the Whova app!

STAY IN THE KNOW

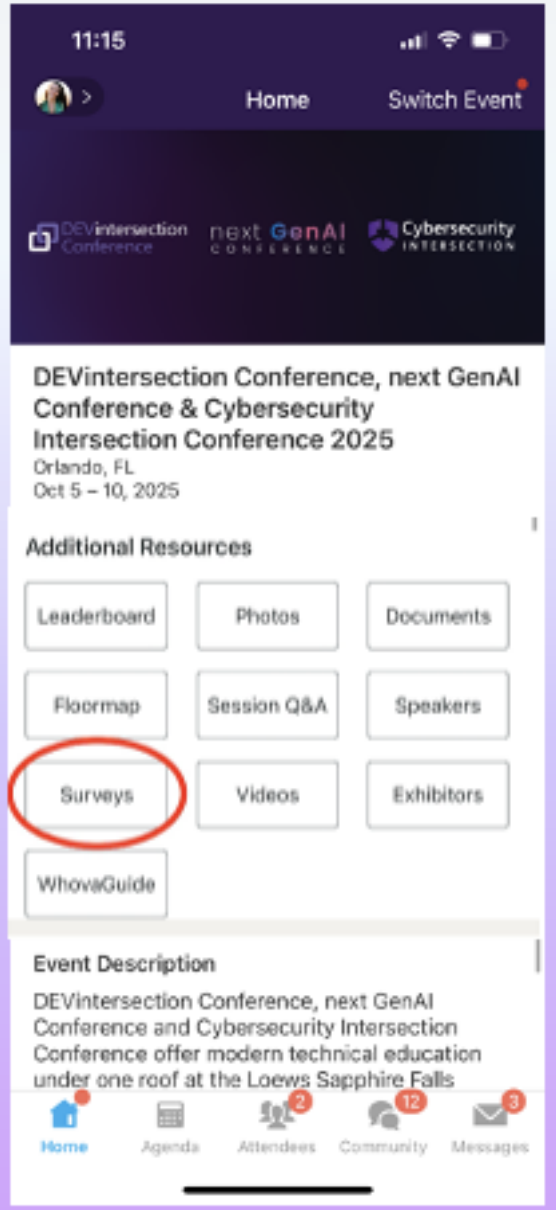
- Receive event updates and notifications
- Connect with other attendees
- Personalize your agenda
- Network & join meet-ups
- Access to Event Documents

Having Trouble Joining?

Search for our event, **DEVintersection Conference, next GenAI Conference & Cybersecurity Intersection Conference 2025.**

Event Invitation Code: ORLANDO2025

Session Feedback Surveys



We really want to hear from YOU!

In the pursuit of making next year's Conference even better, we want to hear your feedback about this session.

Here's How -

- *Simply go to the Whova App on your smartphone*
- *Scroll down on the Homepage to 'Additional Resources' to click "Surveys'.*
- *Click Session Feedback.*
- *Scroll down to find this session title.*
- *Complete the session feedback survey.*
- *Finally, click 'Submit'*

It's just that easy!

Audience

- Anyone using GitHub Actions
- Anyone thinking about moving to GitHub Actions (more on this later)
- People interested in DevOps but rarely/never get to do it

Agenda

- What is CI and the two CDs?
- Things every CI/CD workflow should have
- What are GitHub Actions
- GitHub Actions concepts
- How to share GitHub Actions across the enterprise
- Configuring Optimal GitHub Settings
- Practical GitHub Actions Tips

Goals

- Understand what GitHub Actions are
- Takeaways even if you’re experienced with GitHub Actions

Who am I?

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

Poll

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

Let’s talk
CI + CD + CD first

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

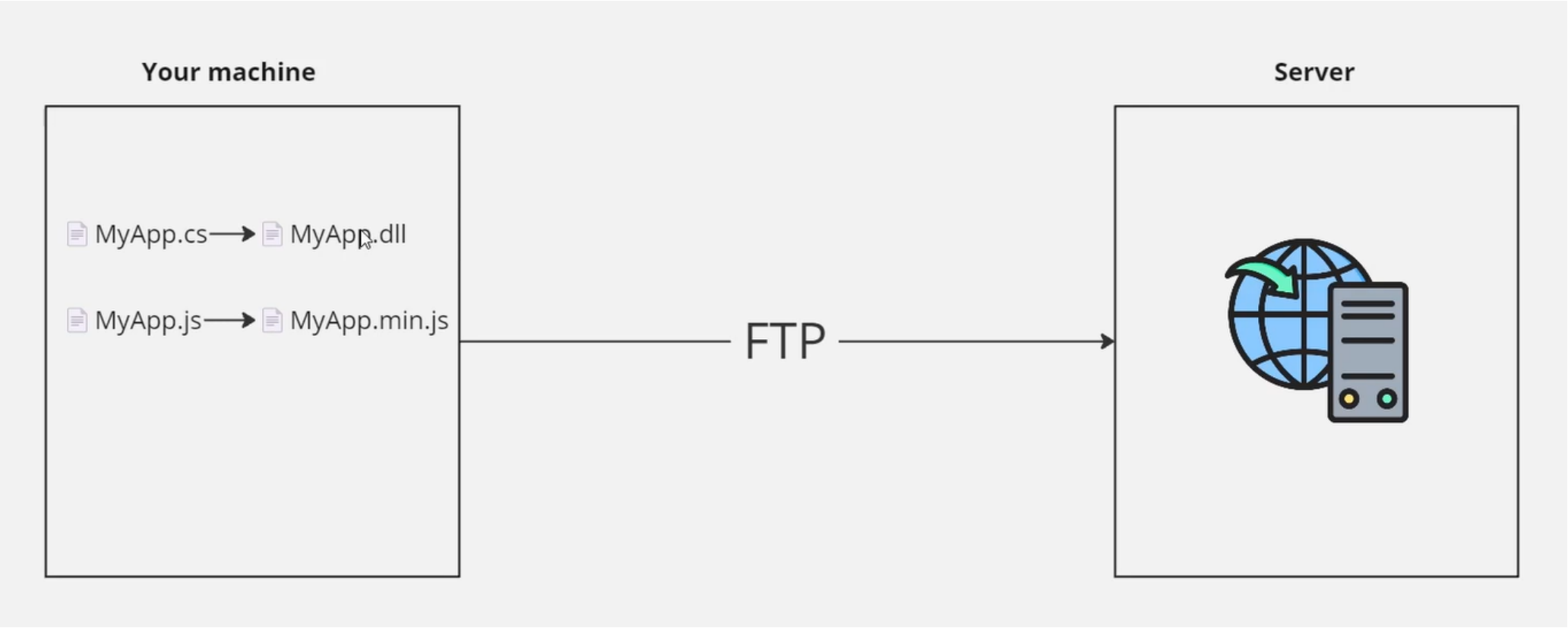
What is Continuous Deployment?

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

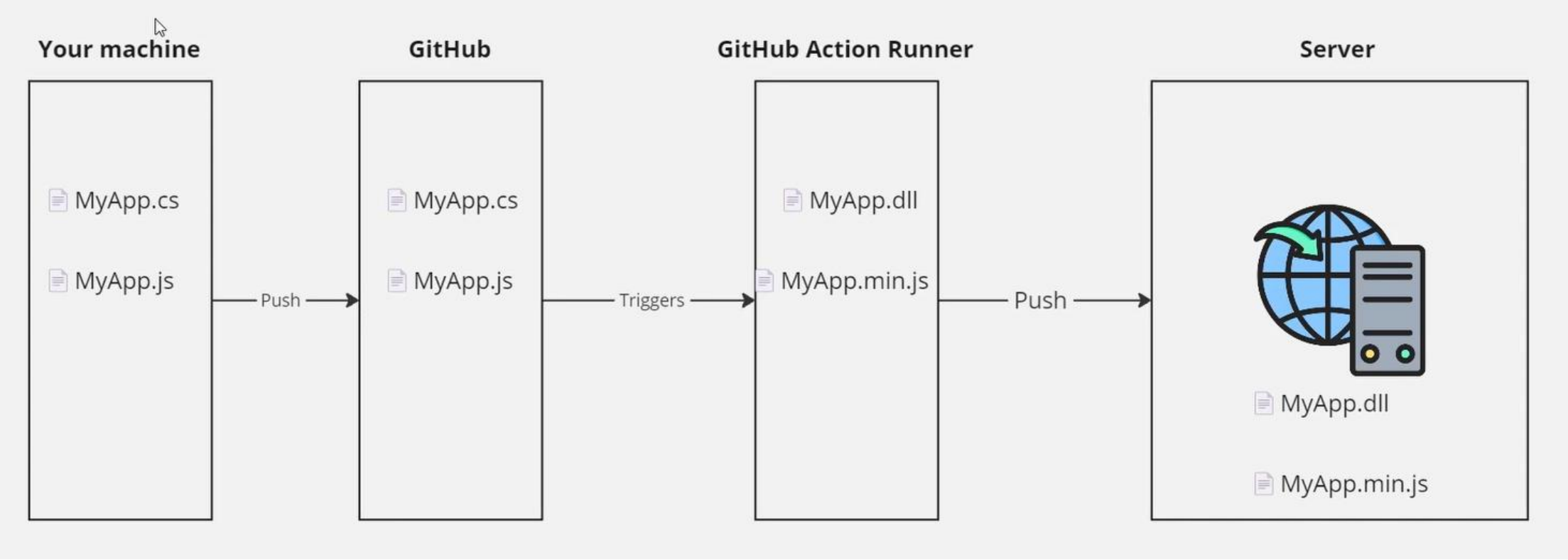
Why CI/CD?

- 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

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

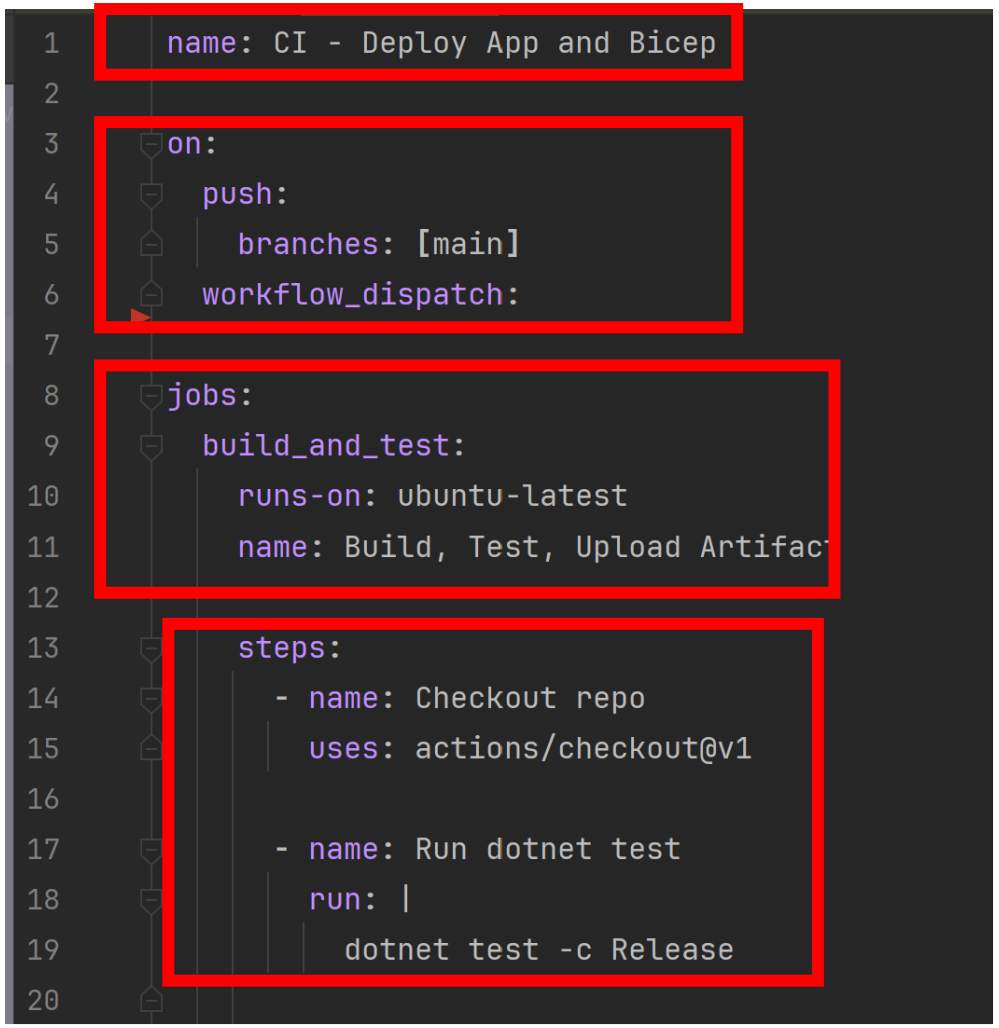
What are GitHub Actions?

- 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
- Written using YAML
  - Use a GHA Extension in your editor
- ~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

```

Live Demo

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)

Optimal GitHub settings

- Ruleset
- Enforce PRs
- Enforce Reviewer Count
- Enforce Status Check must pass

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
```

Consume reusable workflow

```yaml
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
```

Reusable Workflows In Another Repository

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

```

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"
```

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

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

“But I’m on Azure DevOps”

- Azure DevOps is NOT going away anytime soon
- Azure DevOps is NOT going away anytime soon
- The strategic long term investment is in GitHub
- GitHub has thousands of people workin on it, Azure DevOps has a couple hundred
- Quietly Microsoft sales folks are pushing people to GitHub
- Microsoft Build had 1 session on AzDO, GitHub referenced 67 times
- And the AzDO session was - **Making Azure DevOps and GitHub Greater than the Sum of their Parts**

Resources

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

Questions?

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

Add me on LinkedIn:

Slides at scottsauber.com

Thanks!

Slides at scottsauber.com

UPCOMING 2025 & 2026 EVENTS

Power Platform COMMUNITY CONFERENCE

OCTOBER 28 - 31, 2025

WORKSHOPS OCT 26, 27 & 31

MGM GRAND LAS VEGAS

Microsoft 365 COMMUNITY CONFERENCE

APRIL 21 - 23, 2026

WORKSHOPS APRIL 19, 20 & 24

UNIVERSAL ORLANDO, FL