Better Code with CI

Pete Freitag, July 2022

foundeo inc.

Pete Freitag

About Me

- Decades of ColdFusion Experience, stuff I've built:
  - Wrote ColdFusion Lockdown Guides for CF9-CF2021
  - Built cfscript.me and cfdocs.org
  - Stay Updated: petefreitag.com, twitter.com/pfreitag, cfbreak.com, cfml slack
  - My Company: **Foundeo Inc.** provides consulting, tools and services for CF Developers focused on security: Fixinator, FuseGuard, HackMyCF

foundeo inc.

Agenda

- Motivation for CI
- Hello World Examples
- Actual Useful Examples
  - We'll look at various CI features as we go through the examples

Late One Friday Afternoon....

You made a simple code fix.

Photo by nubelson-fernandes on unsplash

So simple of a fix...

It was only one line of code!

Photo by krakenimages on unsplash

Did I mention it was Friday....

Afternoon

Photo by nubelson-fernandes on unsplash

Guess what happened

Syntax Error

- cfqueryparma is not a valid tag
- We could have detected this problem before the developer went home using CI.

Photo by Photo by Karolina Kołodziejczak on unsplash

What is CI?

Or continuous integration?

"Continuous integration is a software development practice where members of a team use a version control system and frequently integrate their work to the same location, such as a main branch. Each change is built and verified to detect integration errors as quickly as possible."

Via: AWS CodePipeline User Guide

What is CI?

“A script that runs on an event”

What is CD?

Or continuous delivery? Or continuous deployment?

"Continuous integration is focused on automatically building and testing code, as compared to *continuous delivery*, which automates the entire software release process up to production."

Via: AWS CodePipeline User Guide

We aren't going to cover CD, but you will often see the term CI/CD together.

Requirements

For implementing continuous integration

- Code
- CI Software / Service

Version Control

Works best

- **Version Control Not Required** - just because you can doesn't mean you should
- **Need Version Control?** GitHub, GitLab, Bitbucket all provide free version control with builtin CI tools
- **Git Not Required** - most resources assume you are using git

"Version Control is not a fad."
-Pete Freitag

What are some CI tools?

GitLab

GitHub

Bitbucket

circleci

Jenkins

CI Tools

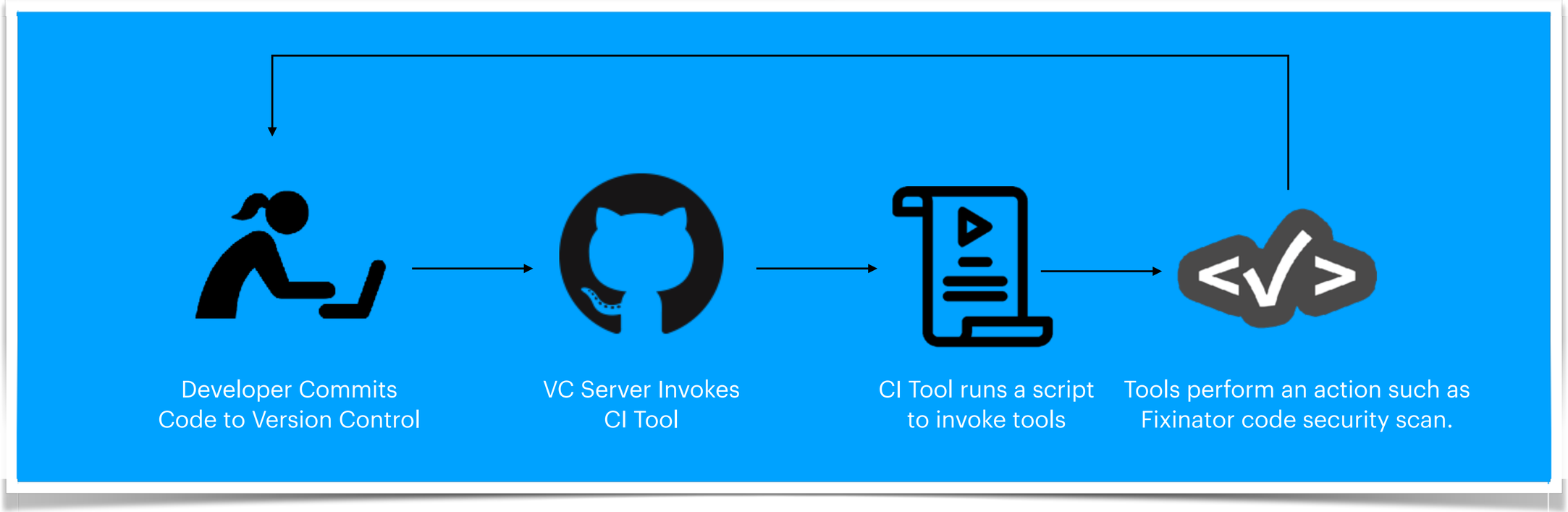
There is no shortage

- Github Actions
- GitLab Pipelines
- Bitbucket Pipelines
- CircleCI
- Azure DevOps
- AWS CodeBuild
- Jenkins
- TravisCI
- TeamCity
- Buddy

Some are easier / better than others

What does the CI Server do?

- Waits for an event: commit code, push a button, schedule
- Executes a script based upon that event



script by Berkah Icon from the Noun Project

What does a CI Script Look Like?

Most use YAML

```yaml
on:
  push:
    branches: [ main ]

jobs:
  fixinator:
    runs-on: ubuntu-latest
    container:
      image: ghcr.io/foundeo/fixinator-docker/fixinator-docker:latest

    steps:
    - uses: actions/checkout@v2
    - name: Run Fixinator
      run: box fixinator
      env:
        FIXINATOR_API_KEY: ${{ secrets.FIXINATOR_API_KEY }}
```

YAML Time

YAML Ain't Markup Language™

- YAML is a “human-friendly” data serialization language for all programming languages.
  - Similar to JSON

```
{
    "fruit": ["Apple", "Orange"]
}
```

JSON

==

```
fruit:
  - Apple
  - Orange
```

YAML

More YAML

- Indentation matters in YAML
  - This will be your first YAML mistake!
  - Tabs are forbidden, use 2 or 4 spaces
- Unlike JSON, YAML supports comments
- You don't need to be a YAML expert

```
#YAY FOR COMMENTS
fruit:
  - Apple
  - Orange
```

Different Syntax

For Different Folks

```
image: java:8

hello:
  script:
    - echo HelloWorld
```

GitLab

```
on:
  push:
    branches: [ main ]

jobs:
  hello:
    runs-on: ubuntu-latest
    steps:
    - name: Say Hello
      run: echo Hello
```

GitHub

```
image: openjdk:8

pipelines:
  default:
    - step:
        script:
          - echo HelloWorld
```

Bitbucket

Where does the yaml go?

Commit a file to your repository named...

- GitHub Actions: /.github/workflows/whatever.yml
- BitBucket: /bitbucket-pipelines.yml
- GitLab: /.gitlab-ci.yml
- Azure DevOps: /azure-pipeline.yml
- AWS CodeBuild: /buildspec.yml
- Jenkins: Jenkinsfile
- CircleCI: /.circleci/config.yml

We will use GitHub Actions Syntax

Everything you will see is possible on any CI platform, but the syntax will differ

All examples are here: https://github.com/foundeo/cfml-ci-examples

A Simple CI Script

For GitHub Actions

```
on:
  push:
    branches: [ main ]

jobs:
  hello:
    runs-on: ubuntu-latest
    steps:
    - name: Say Hello
      run: echo Hello
```

A Simple CI Script

For GitHub Actions

```
on:
  push:
    branches: [ main ]

jobs:
  hello:
    runs-on: ubuntu-latest
    steps:
    - name: Say Hello
      run: echo Hello
```

When you push
code to the main branch

A Simple CI Script

For GitHub Actions

```
on:
  push:
    branches: [ main ]

jobs:
  hello:
    runs-on: ubuntu-latest
    steps:
    - name: Say Hello
      run: echo Hello
```

Start a job called hello running on the latest version of ubuntu linux

A Simple CI Script

For GitHub Actions

```
on:
  push:
    branches: [ main ]

jobs:
  hello:
    runs-on: ubuntu-latest
    steps:
    - name: Say Hello
      run: echo Hello
```

In the job
step “Say Hello”
Run the
command: echo Hello

Jobs

Github Actions

- You can define multiple jobs
  - Jobs run in parallel, unless dependencies are defined
- Jobs consist of one or more steps

Run

- A bat file
- A shell script
- Anything executable

Add CI to your Project

Two Phases

Phase 1

- Low effort, high value
- Get up an running in a few hours

Phase 2

- Takes effort: days, weeks
- Write tests
- Allows you to deploy changes with confidence.

Doing Something Useful

Phase 1: The low hanging fruit

- Start with some easy first steps that you can duplicate for every CF project
  - **Does it compile?** Or do we have syntax errors?

Does it compile?

ColdFusion Example

```
on:
  push:
    branches: [ main ]

jobs:
  cfcompile:
    runs-on: ubuntu-latest
    container:
      image: adobecoldfusion/coldfusion2021:2021.0.4
    env:
      acceptEULA: YES
    steps:
      - uses: actions/checkout@v3
      - name: Compile CFML
        run: /opt/coldfusion/cfusion/bin/cfcompile.sh -cfruntimeuser root -webroot $GITHUB_WORKSPACE -dir $GITHUB_WORKSPACE
```

Does it compile?

ColdFusion Example

Pull the Adobe ColdFusion Docker image For CF2021 update 4

```
on:
  push:
    branches: [ main ]

jobs:
  cfcompile:
    runs-on: ubuntu-latest
    container:
      image: adobecoldfusion/coldfusion2021:2021.0.4
    env:
      acceptEULA: YES
    steps:
      - uses: actions/checkout@v3
      - name: Compile CFML
        run: /opt/coldfusion/cfusion/bin/cfcompile.sh -cfruntimeuser root -webroot $GITHUB_WORKSPACE -dir $GITHUB_WORKSPACE
```

Does it compile?

ColdFusion Example

Pass environment Variables into the docker container

```
on:
  push:
    branches: [ main ]

jobs:
  cfcompile:
    runs-on: ubuntu-latest
    container:
      image: adobecoldfusion/coldfusion2021:2021.0.4
    env:
      acceptEULA: YES
    steps:
      - uses: actions/checkout@v3
      - name: Compile CFML
        run: /opt/coldfusion/cfusion/bin/cfcompile.sh -cfruntimeuser root -webroot $GITHUB_WORKSPACE -dir $GITHUB_WORKSPACE
```

Does it compile?

ColdFusion Example

```
on:
  push:
    branches: [ main ]

jobs:
  cfcompile:
    runs-on: ubuntu-latest
    container:
      image: adobecoldfusion/coldfusion2021:2021.0.4
    env:
      acceptEULA: YES
    steps:
      - uses: actions/checkout@v3
      - name: Compile CFML
        run: /opt/coldfusion/cfusion/bin/cfcompile.sh -cfruntimeuser root -webroot $GITHUB_WORKSPACE -dir $GITHUB_WORKSPACE
```

Checkout the code that was pushed

Does it compile?

ColdFusion Example

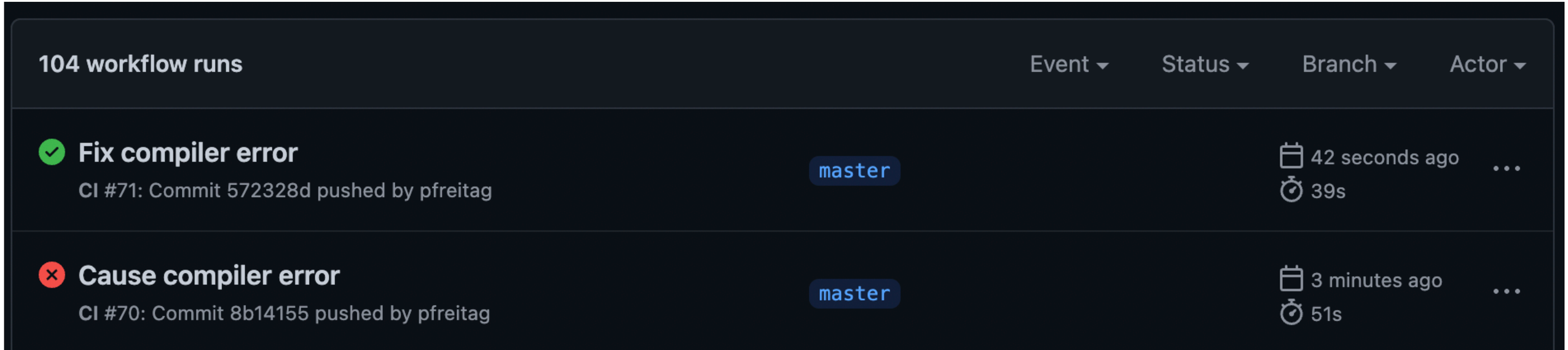
```
on:
  push:
    branches: [ main ]

jobs:
  cfcompile:
    runs-on: ubuntu-latest
    container:
      image: adobecoldfusion/coldfusion2021:2021.0.4
    env:
      acceptEULA: YES
    steps:
      - uses: actions/checkout@v3
      - name: Compile CFML
        run: /opt/coldfusion/cfusion/bin/cfcompile.sh -cfruntimeuser root -webroot $GITHUB_WORKSPACE -dir $GITHUB_WORKSPACE
```

Run cfcompile against your code

Build Status

Each Job can Pass or Fail

- When a program runs it returns an exit code:
  - 1 = error
  - 0 = success
- The cfcompile tool will return an exit code of 1 when it fails to compile the code



Compile Example

Finding Security Vulnerabilities

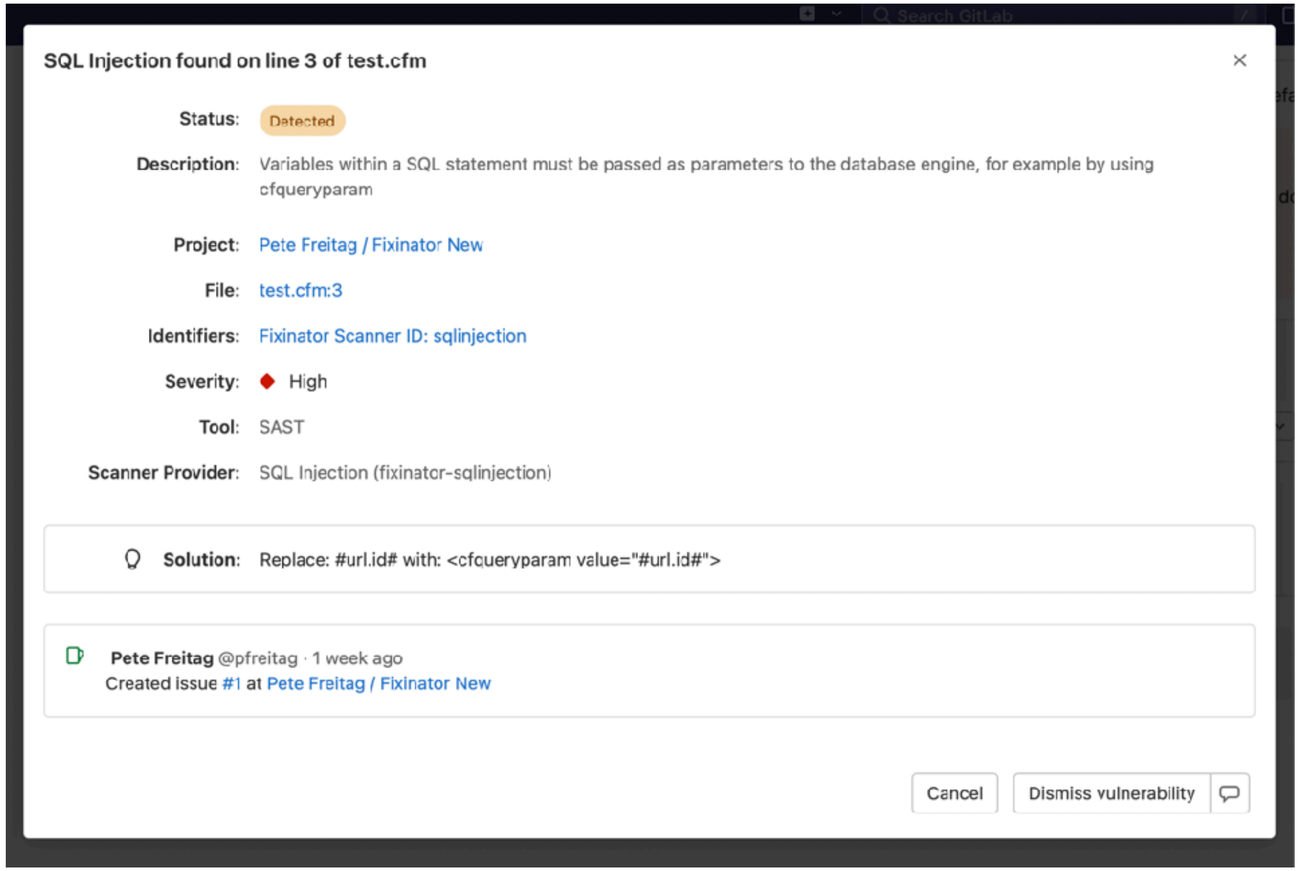
CFML Static Code Analysis

- Fixinator - commercial CFML code security scanner
  - Focused on ColdFusion
  - Maintained / Updated Frequently
  - Highly Configurable
  - Designed to run inside CI
  - Trial: fixinator.app/try/

Fixinator CI Examples

Guides written for all major CI platforms

- Getting Started Guides: https://github.com/foundeo/fixinator/wiki/Continuous-Integration-Guide
  - Github, GitLab, Bitbucket, Jenkins, Azure DevOps, AWS CodeBuild, TeamCity, CircleCI, and TravisCI
- Security Finding Integration on some platforms.



Example GitLab Security Report UI

Fixinator Example

Fixinator GitHub Action

Fixinator

More than SQL Injection

- Detects several known CFM backdoors, web shells
- Vulnerable CFML Dependencies (via forgebox box.json)
- Detects several vulnerable jar files
- CFML Coding Vulnerabilities:
    - SQL Injection, XSS, Remote Code Execution, LDAP Inject, Unsafe File Access, ReDOS, Path Traversal, Weak Session Cookie Settings, Insecure Application Settings, Weak Hash / Encryption Algorithms, and more.

Fixinator

Super Configurable

- Ignore Scanners
- Ignore Variable Patterns
  - For example if you created myOwnAwesomeHTMLEncoder() XSS scanner can ignore variables wrapped in it.
  - Or #application.table_prefix# - ignored by SQL Injection
- Ignore code, special comment <!—- ignore: sql-injection comment —->
- Scan only changed files

Test Quality / Standards

May Take Some Config

- Additional Code Scanners
  - CodeChecker - supports a .codechecker.json file
  - CFLint - define config in a .cflintrc file

Other Trigger Events

Schedule

```
on:
  schedule:
    - cron: '30 9 1 * *'
```

Run at 9:30 UTC on the First of every month

Other Trigger Events

Manual

```
on:
  workflow_dispatch:
```

Other Trigger Events

Manual with Inputs

```
on:
  workflow_dispatch:
    inputs:
      your_name:
        description: 'What is your name?'
        default: 'Pete'
        required: true
```

Referenced as: ${{ github.event.inputs.your_name }}

Other Trigger Events

Match Pushed Tag

```
on:
  push:
    tags:
      - v*
```

Runs whenever a tag is pushed starting with v

Secrets

Leverage CI Secret Manager

- Each CI Platform provides secure secret storage.
  - In Github Actions Organization or Repository level secrets can be defined
- The CI platform also makes sure the secret value is not output in the build logs
- Secret is not available to the job / step unless passed to it (eg via env)
- Use this for API keys, passwords, etc.

```
env:
  FIXINATOR_API_KEY: ${{ secrets.FIXINATOR_API_KEY }}
```

Matrix

```
jobs:
  job_name:
    strategy:
      matrix:
        cfengine: ["adobe@2021", "adobe@2018"]
        java: ["openjdk11", "openjdk8"]
```

Job Dependency

```
jobs:
  test:
   #…
  build:
    needs: [test]
```

Runs On

```
jobs:
  job_name:
    runs-on: ubuntu-latest
```

Choosing GitHub-hosted runners

If you use a GitHub-hosted runner, each job runs in a fresh instance of a virtual environment specified by `runs-on`.

Available GitHub-hosted runner types are:

Virtual environment	YAML workflow label	Notes
Windows Server 2022	`windows-latest` or `windows-2022`	The `windows-latest` label currently uses the Windows Server 2022 runner image.
Windows Server 2019	`windows-2019`	
Ubuntu 22.04	`ubuntu-22.04`	Ubuntu 22.04 is currently in public beta.
Ubuntu 20.04	`ubuntu-latest` or `ubuntu-20.04`	
Ubuntu 18.04	`ubuntu-18.04`	
macOS Monterey 12	`macos-12`	
macOS Big Sur 11	`macos-latest` or `macos-11`	The `macos-latest` label currently uses the macOS 11 runner image.
macOS Catalina 10.15	`macos-10.15`	

Note: The `-latest` virtual environments are the latest stable images that GitHub provides, and might not be the most recent version of the operating system available from the operating system vendor.

https://docs.github.com/en/actions/using-workflows/workflow-syntax-for-github-actions#jobsjob_idruns-on

Permissions

```yaml
permissions:
  contents: read
jobs:
  publish_package:
    permissions:
      packages: write
```

Available scopes and access values:

```yaml
permissions:
  actions: read|write|none
  checks: read|write|none
  contents: read|write|none
  deployments: read|write|none
  id-token: read|write|none
  issues: read|write|none
  discussions: read|write|none
  packages: read|write|none
  pages: read|write|none
  pull-requests: read|write|none
  repository-projects: read|write|none
  security-events: read|write|none
  statuses: read|write|none
```

If you specify the access for any of these scopes, all of those that are not specified are set to `none`.

You can use the following syntax to define read or write access for all of the available scopes:

```yaml
permissions: read-all|write-all
```

You can use the following syntax to disable permissions for all of the available scopes:

```yaml
permissions: {}
```

https://docs.github.com/en/actions/using-workflows/workflow-syntax-for-github-actions#permissions

Writing Your own CI Tools

Using CommandBox Task Runners

- box task run taskFile=path/to/some.cfc

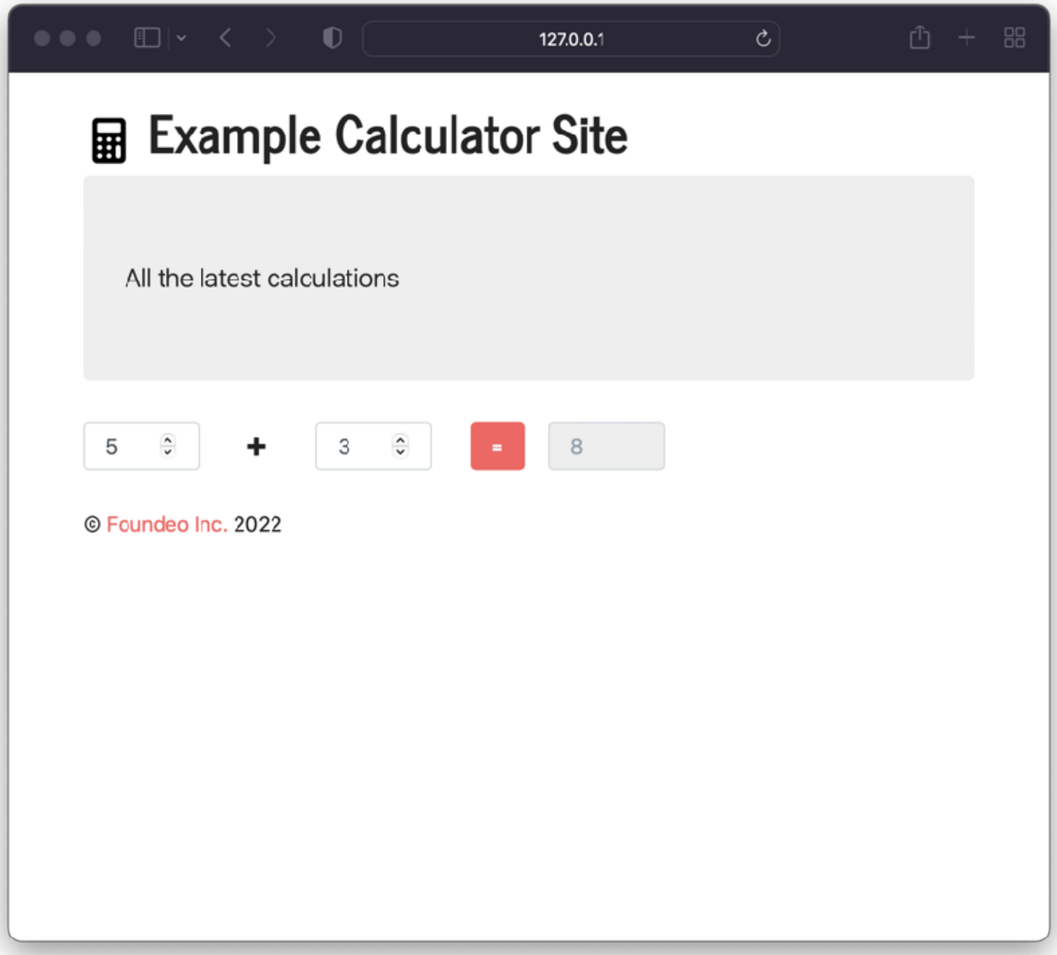
```
component {

    function run() {
        var dayName = dayOfWeekAsString(dayOfWeek(now()));
        if ( dayName == "Friday") {
            error("Sorry no changes on Friday!")
        } else {
            print.greenLine("Ok, it is #dayName# - all good!");
        }
    }

}
```

Phase 2: Functional Testing

Running and testing your app inside the CI runner

- Start an instance of your app using CommandBox or Docker
  - Test via curl (http request)
  - Test via TestBox
  - Test API via PostMan
  - Test front end test Puppeteer
  - And many other tools

Our Example App



Our Mission Critical Code

```
component {

    remote function add(x, y) {
        return x+y;
    }

}
```

Functional Test Examples

With exceptional automated testing in place, you actually can deploy on Friday!

Just because you can, doesn’t mean you should

We’ve only scratched the surface

Much more is possible with CI, keep learning

Photo by alessandro-erbetta on unsplash

https://fixinator.app/try/

Thank You!

Questions?

foundeo.com/contact/

foundeo inc.

https://fixinator.app/try/

Thank You!

Questions?

Contact Me: pete@foundeo.com

Adobe Gift Card Question:

What does my last name “Freitag” mean in German?

foundeo inc.