GitHub Actions

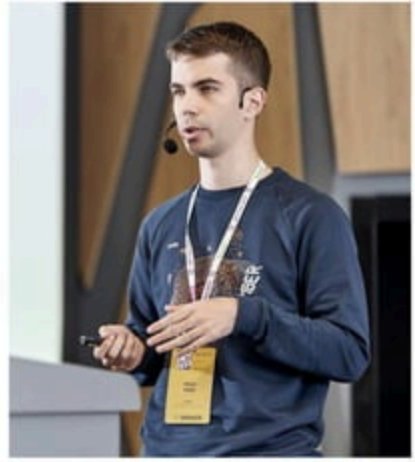
...in action!

Speaker: Alexey Golub

@Tyrrrz

/whois ${speaker}

- Open-source developer
- Conference speaker & blogger
- C#, F#, JavaScript
- Cloud & web
- Automation & DevOps



Speaker: **Alexey Golub**

@Tyrrrz

GitHub Actions at a glance

- Globally available since November 13, 2019
- Based on Azure Pipelines
- Native integration with GitHub API
- YAML-based configuration
- Modular architecture & community-driven
- Runners on Windows, Linux, macOS, or self-hosted
- Available with Free, Pro, Team, Enterprise Cloud, One

Public repositories

Free! 

Private repositories

Product	Storage	Minutes (monthly)
GitHub Free	500 MB	2,000
GitHub Pro	1 GB	3,000
GitHub Team	2 GB	10,000
GitHub Enterprise Cloud	50 GB	50,000

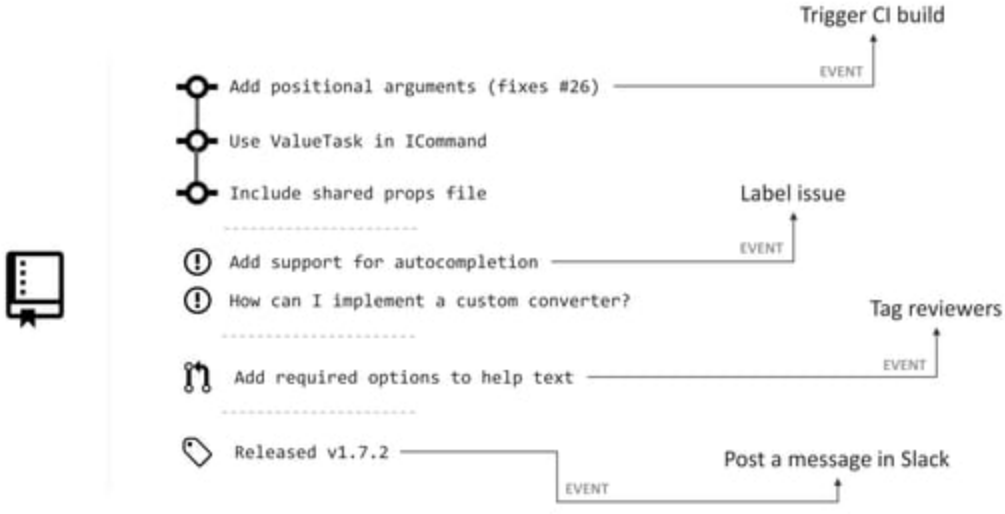
Operating system	Minute multiplier
Windows	2
Linux	1
macOS	10

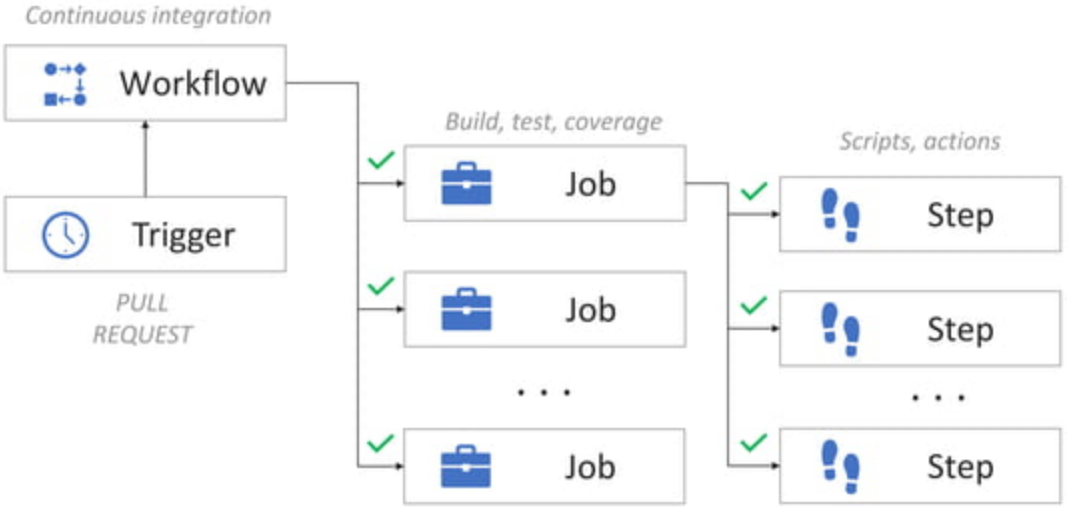
Operating system	Per-minute rate
Windows	$0.016
Linux	$0.008
macOS	$0.080

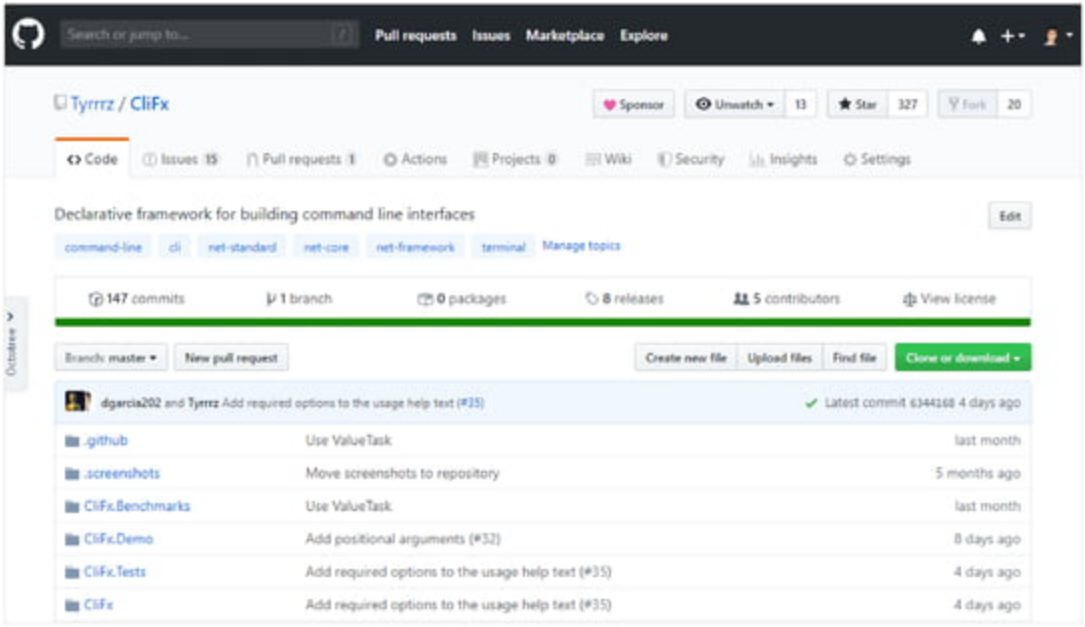
GitHub Actions

=

IFTTT for GitHub repositories


Trigger CI build
EVENT
Add positional arguments (fixes #26)
Use ValueTask in ICommand
Include shared props file
Label issue
EVENT
Add support for autocompletion
How can I implement a custom converter?
Tag reviewers
EVENT
Add required options to help text
Released v1.7.2
EVENT
Post a message in Slack


Continuous integration
Workflow
Trigger
PULL
REQUEST
Build, test, coverage
Job
Job
Job
Scripts, actions
Step
Step
Step



Speaker: **Alexey Golub**

@Tyrrrz

~/.github/workflows/CI.yml

```yaml
name: CI
on: [push, pull_request]

jobs:
  build:
    runs-on: windows-latest

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

    - name: Install .NET Core
      uses: actions/setup-dotnet@v1
      with:
        dotnet-version: 3.1.100

    - name: Build & test
      run: dotnet test --configuration Release

    - name: Build & publish
      run: dotnet publish LightBulb/ -o LightBulb/bin/Publish/ --configuration Release

    - name: Upload build artifacts
      uses: actions/upload-artifact@master
      with:
        name: LightBulb
        path: LightBulb/bin/Publish/
```

Trigger on new commits and pull requests

Clone repository and checkout HEAD
(commit hash is passed as env var)

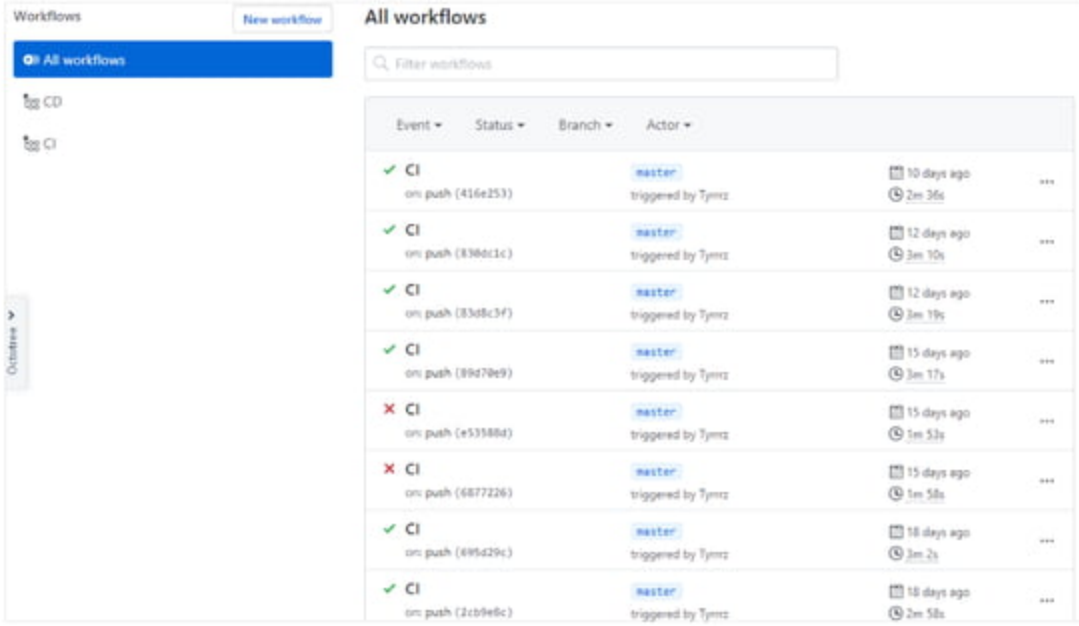
Install .NET Core v3.1.100

Run custom shell scripts

Upload specified directory as
a ZIP artifact

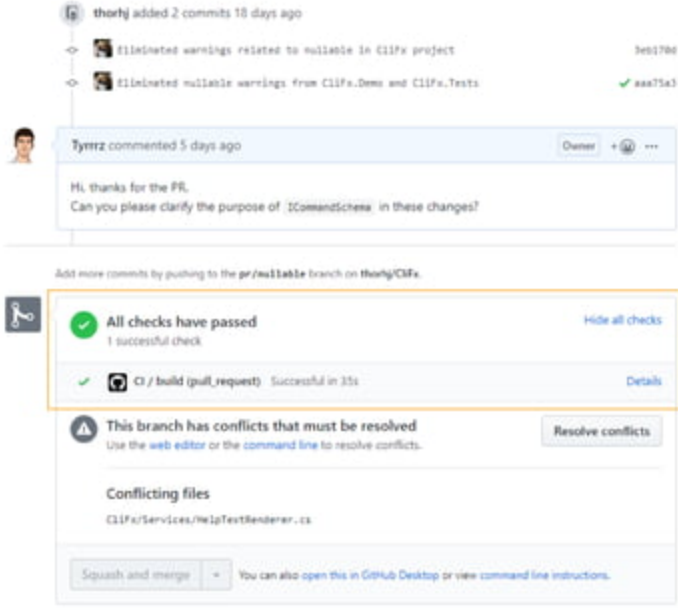
Speaker: Alexey Golub

@Tyrrrz



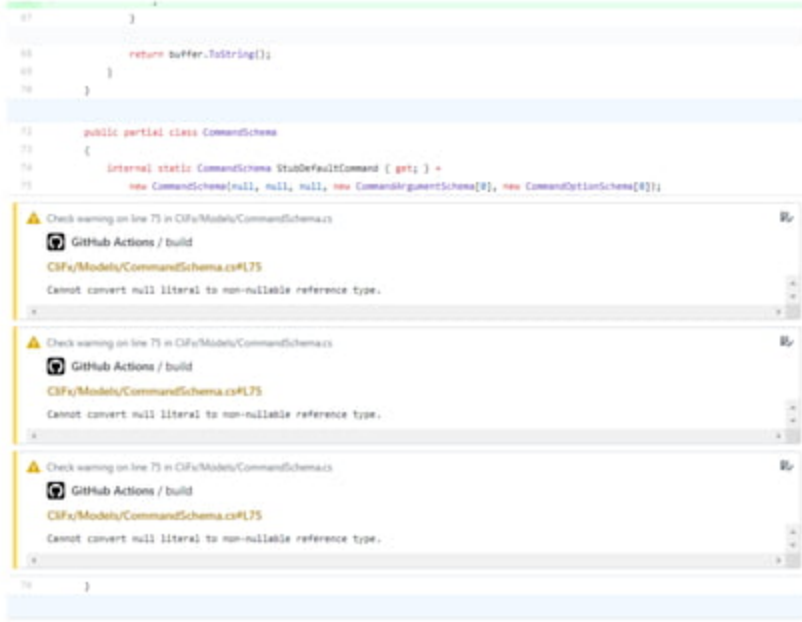
Speaker: **Alexey Golub**

@Tyrrrz



Speaker: Alexey Golub

@Tyrrrz


}
return buffer.ToString();
}
}
public partial class CommandSchema
{
internal static CommandSchema StubDefaultCommand { get; } =
new CommandSchema(null, null, null, new CommandArgumentSchema[0], new CommandOptionSchema[0]);
Check warning on line 75 in CliFx/Models/CommandSchema.cs
GitHub Actions / build
CliFx/Models/CommandSchema.cs#L75
Cannot convert null literal to non-nullable reference type.
Check warning on line 75 in CliFx/Models/CommandSchema.cs
GitHub Actions / build
CliFx/Models/CommandSchema.cs#L75
Cannot convert null literal to non-nullable reference type.
Check warning on line 75 in CliFx/Models/CommandSchema.cs
GitHub Actions / build
CliFx/Models/CommandSchema.cs#L75
Cannot convert null literal to non-nullable reference type.
}

Triggers

- **GitHub API events** ⚡
  `push`, `pull_request`, `issues`, `release`, and 20 others
- **Schedule** 🕒
  Cron syntax, e.g.: `*/15 * * * *`
- **Manual** 🌐
  POST to /repos/:owner/:repo/dispatches

```yaml
# Trigger on push events on s
pecific branches
on:
  push:
    branches:
      - 'master'
      - 'release/*'
```

```yaml
# Trigger on manual dispatch
on: repository_dispatch
```

```yaml
# Trigger every midnight UTC
on:
  schedule:
    - cron: '0 0 * * *'
```

```yaml
# Trigger when an issue is opened o
r labeled
on:
  issues:
    types: [opened, labeled]
```

Referencing actions

- By GitHub repository 

  `{owner}/{repo}@{ref}` — `jessfraz/branch-cleanup-action@master`

  `{owner}/{repo}/{path}@{ref}` — `johndoe/my-actions/push-image@v1`

- By file path 

  `./path/to/dir` — `./.github/actions/my-action`

- By Docker image 

  `docker://{image}:{tag}` — `docker://hello-world:latest`

Advanced configurations

Matrices

Ubuntu-16.04	Ubuntu-18.04
Node v6	Node v6
Node v8	Node v8
Node v10	Node v10

```yaml
name: Matrix Reloaded

on: [push]

jobs:
  build:
    runs-on: ${{ matrix.os }}

    strategy:
      max-parallel: 4
      matrix:
        os: [ubuntu-16.04, ubuntu-18.04]
        node-ver: [6, 8, 10]

    steps:
      - uses: actions/setup-node@v1
        with:
          node-version: ${{ matrix.node-ver }}
```

Reference variables from the matrix

Docker containers

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

    services:
      redis:
        image: redis
        ports:
        - 6379/tcp
        options: --entrypoint redis-server

    steps:
      - uses: actions/checkout@v1

      - run: node client.js
        env:
          REDIS_HOST: localhost
          REDIS_PORT: ${{ job.services.redis.ports[6379] }}
```

Image from the Docker registry

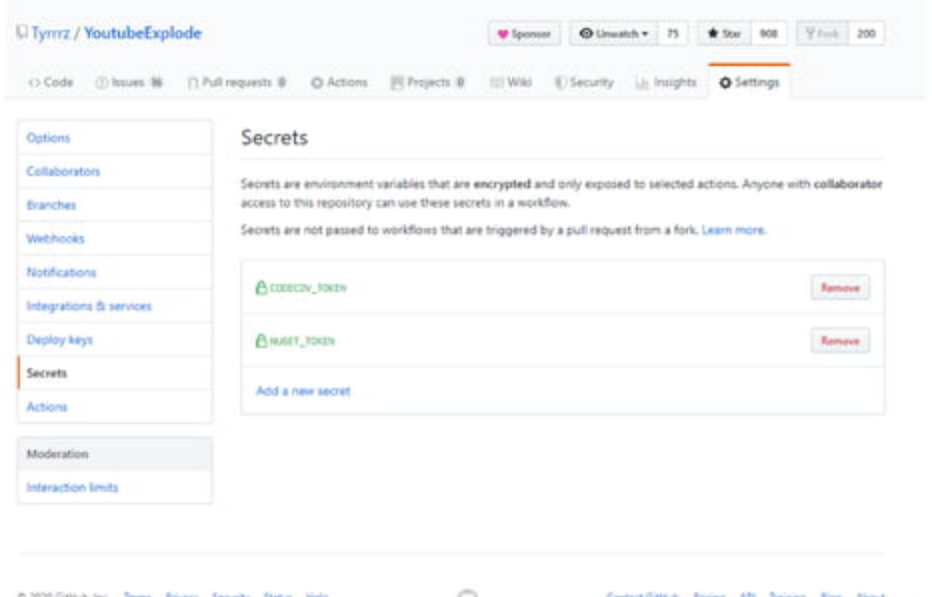
Bind port 6379 in container to a random port on host

Custom arguments passed to `docker create`

Exposed port is resolved dynamically

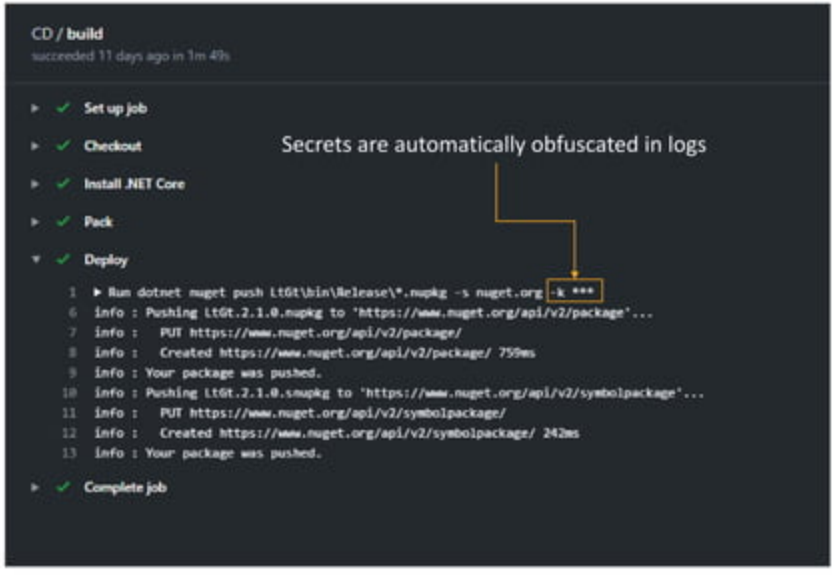
Secrets

```yaml
steps:

# ...

- name: Collect coverage report
  run: |
    choco install codecov --no-progress
    codecov -f LtGt.Tests/bin/Release/Coverage.xml -t ${{secrets.CODECOV_TOKEN}}
```



Speaker: **Alexey Golub**

@Tyrrrz


CD / build
succeeded 11 days ago in 1m 49s
Set up job
Checkout
Install .NET Core
Pack
Deploy
Secrets are automatically obfuscated in logs
1 ▶ Run dotnet nuget push LtGt\bin\Release*.nupkg -s nuget.org -k ***
6 info : Pushing LtGt.2.1.0.nupkg to 'https://www.nuget.org/api/v2/package'...
7 info : PUT https://www.nuget.org/api/v2/package/
8 info : Created https://www.nuget.org/api/v2/package/ 759ms
9 info : Your package was pushed.
10 info : Pushing LtGt.2.1.0.snupkg to 'https://www.nuget.org/api/v2/symbolpackage'...
11 info : PUT https://www.nuget.org/api/v2/symbolpackage/
12 info : Created https://www.nuget.org/api/v2/symbolpackage/ 242ms
13 info : Your package was pushed.
Complete job

Action to action I/O

```yaml
- name: Create release
  id: create_release
  uses: actions/create-release@v1
  env:
    GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
  with:
    tag_name: ${{ github.ref }}
    release_name: ${{ github.ref }}

- name: Upload release asset
  uses: actions/upload-release-asset@v1.0.1
  env:
    GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
  with:
    upload_url: ${{ steps.create_release.outputs.upload_url }}
    asset_path: DiscordChatExporter/bin/Publish/Archive.zip
    asset_name: DiscordChatExporter.zip
```

Actions can be referenced by their ID

Resolved from the outputs of another action

Conditionals

```yaml
jobs:
  build:
    runs-on: ubuntu-18.04

    steps:
    - uses: actions/checkout@v1
    - uses: actions/setup-dotnet@v1
      with:
        dotnet-version: 3.1.100

    - run: dotnet test src
    - run: dotnet pack src

    - if: github.event_name == 'push' && startsWith(github.ref, 'refs/tags/v')
      run: dotnet nuget push src/**.nupkg -k ${{secrets.NUGET_TOKEN}}
```

Conditional expression

Things we can do with GitHub Actions

- Run tests on every commit
- Publish Docker image when a tag is pushed
- Label an issue by content when it's created
- Run nightly E2E tests
- Automatically close stale issues every week
- Invite new contributors to sign the CLA when a PR is opened
- Automatically format code on push
- ...

Examples of actions you can use in your workflows

actions/github-script

```yaml
on:
  issues: {types: opened}

jobs:
  comment:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/github-script@0.4.0
        with:
          github-token: ${{secrets.GITHUB_TOKEN}}
          script: |
            github.issues.createComment({
              issue_number: context.issue.number,
              owner: context.repo.owner,
              repo: context.repo.repo,
              body: '👋 Thanks for reporting!'
            })
```

Runs inline JS code that uses the GitHub API

actions/stale

```yaml
on:
  schedule:
  - cron: "0 0 * * *"

jobs:
  stale:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/stale@v1
      with:
        repo-token: ${{ secrets.GITHUB_TOKEN }}
        stale-issue-message: 'Issue closed due to inactivity.'
        stale-pr-message: 'PR closed due to inactivity.'
        days-before-stale: 30
        days-before-close: 5
```

Closes stale issues and PRs

sonarsource/sonarcloud-github-action

```yaml
on: push

jobs:
  sonarCloudTrigger:
    name: SonarCloud Trigger
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@master

    - name: SonarCloud Scan
      uses: sonarsource/sonarcloud-github-action@master
      env:
        GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
        SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
```

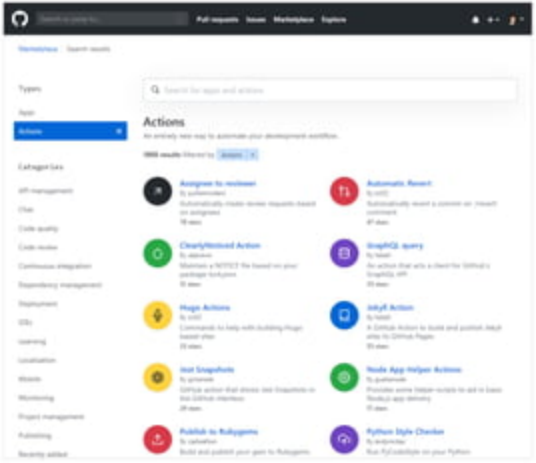
Sends code to SonarCloud to scan for issues

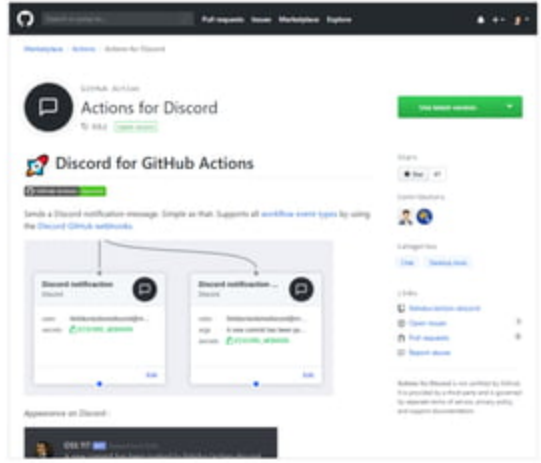
vsoch/pull-request-action

```yaml
on: push

jobs:
  auto-pull-request:
    name: Create pull request for a new branch
    runs-on: ubuntu-latest
    steps:
      - name: pull-request-action
        uses: vsoch/pull-request-action@master
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
          BRANCH_PREFIX: "features/"
          PULL_REQUEST_BRANCH: "master"
```

Automatically creates a pull request when publishing a new branch

Discovering actions





Can’t find an action you need? Make your own!

JavaScript-based actions

- Supported by all runners
- Can only use JavaScript
- Requires a single self-contained .js file as the entry point
- Node modules `@actions/core` & `@actions/github`

~/action.yml

```yaml
name: 'Hello World'
description: 'Greet someone and record time'
inputs:
  who-to-greet:
    description: 'Who to greet'
    required: true
    default: 'World'
outputs:
  time:
    description: 'The time we greeted you'
runs:
  using: 'node12'
  main: 'index.js'
```

Metadata associated with an action

Inputs & outputs

Entry point

~/index.js

```
const core = require('@actions/core');
const github = require('@actions/github');

try {
  // `who-to-greet` input defined in action metadata file
  const nameToGreet = core.getInput('who-to-greet');
  console.log(`Hello ${nameToGreet}!`);
  const time = (new Date()).toTimeString();
  core.setOutput("time", time);

  // Get the JSON webhook payload for the event that triggered the workflow
  const payload = JSON.stringify(github.context.payload, undefined, 2)
  console.log(`The event payload: ${payload}`);

} catch (error) {
  core.setFailed(error.message);
}
```

Docker-based actions

- Supported only by Linux-based runners (for now)
- Can use any language or runtime
- Typically runs slower

~/action.yml

```yaml
name: 'Hello World'
description: 'Greet someone and record time'
inputs:
  who-to-greet:
    description: 'Who to greet'
    required: true
    default: 'World'
outputs:
  time:
    description: 'The time we greeted you'
runs:
  using: 'docker'
  image: 'Dockerfile'
  args:
    - ${{ inputs.who-to-greet }}
```

~/Dockerfile

```
FROM ubuntu:18.04
COPY entrypoint.sh ./
ENTRYPOINT ["/entrypoint.sh"]
```

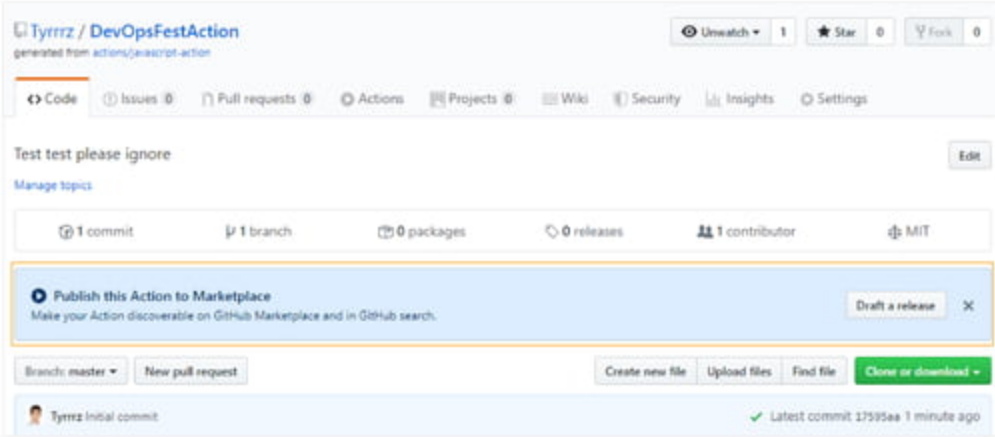
~/entrypoint.sh

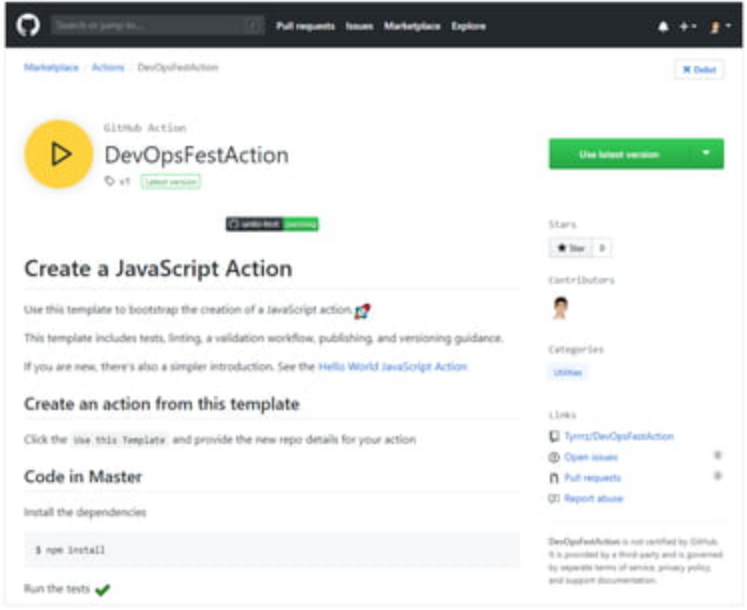
```
#!/bin/sh -l

echo "Hello $1"
time=$(date)
echo ::set-output name=time::$time
```

Special commands

- `::set-output name=action_fruit::strawberry`
- `::set-env name=action_state::yellow`
- `::add-path::/path/to/dir`
- `::debug file=app.js,line=1::Entered method`
- `::warning file=app.js,line=1,col=5::Missing semicolon`
- `::error file=app.js,line=10,col=15::Oops`
- `::add-mask::Mona The Octocat`

Publishing actions to marketplace





Speaker: Alexey Golub

@Tyrrrz

Summary

- GitHub Actions is an automation platform (not just CI/CD)
- Can trigger workflows on various events
- Workflows are based on actions which are sourced by the community
- Free for all public repos, pay-as-you-go for private repos
- Easy to set up and configure to your needs

For the curious

- Awesome Actions by Sarah Drasner
  https://github.com/sdras/awesome-actions
- GitHub Actions Advent Calendar by Edward Thomson
  https://edwardthomson.com/blog/github_actions_advent_calendar.html
- Comprehensive Introduction to GitHub Actions by Tierney Cyren
  https://dev.to/bnb/an-unintentionally-comprehensive-introduction-to-github-actions-ci-blm
- Official documentation
  https://help.github.com/en/actions

Thank you!

https://github.com/Tyrrrz
https://twitter.com/Tyrrrz
https://tyrrrz.me