How to use GitHub Actions with security in mind

SOLIDIFY

https://myoctocat.com

@robbos81

How to use GitHub Actions with security in mind

SOLIDIFY

Rob Bos

DevOps Consultant – Xpirit

The Netherlands

https://myoctocat.com

@robbos81

What are GitHub workflows?

Execute one or more Actions

Workflows triggered by events:

- Push
- Comment
- Creating an Issue
- Release
- Etc.

What are GitHub Actions?

- Steps in the workflows
- Basis: Run a shell script
- Create your own
- Use an existing one from the marketplace

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search GitHub / Sign in

Marketplace / Search results

Types

Apps

Actions ×

Categories

API management

Chat

Code quality

Code review

Continuous integration

Dependency management

Deployment

Search for apps and actions

Sort: Best Match

Actions

An entirely new way to automate your development workflow.

8612 results filtered by Actions ×

Setup Go environment
By actions
Setup a Go environment and add it to the PATH
402 stars

First interaction
By actions
Greet new contributors when they create their first issue or open their first pull request
100 stars

Setup Node.js environment
By actions
Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH
1.1k stars

Upload a Build Artifact
By actions
Upload a build artifact that can be used by subsequent workflow steps
958 stars

Workflow example

main

dotnetcore-webapp / .github / workflows / dotnetcore.yml

```yaml
name: .NET Core

on: [push]

jobs:
  build-and-deploy:
    environment: Production

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v1
    - name: Setup .NET Core
      uses: actions/setup-dotnet@v1
      with:
        dotnet-version: 3.0.100

    # dotnet build
    - name: Build with dotnet
      run: |
        dotnet build --configuration Release ./dotnet-core-webapp/dotnetcore-webapp.csproj
```

GitHub Actions Security

- Repository security
- Runners and security
- Actions and security

- Forking actions
- Keeping up to date

Repository security

- Access to code
- Workflow secrets
- Your code

Code – Who has access?

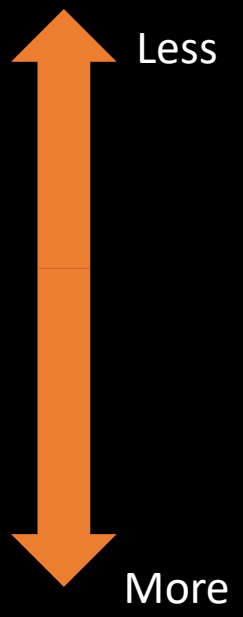
Access levels can be set at:

- Repository
- Organization
- Enterprise

Follow best practices: use teams to group users!

Code – Who has access?

Permission levels



No access
Read only access
Triage: manage issues & pull requests
Write access
Maintain: No sensitive or destructive actions
Admin: full access

Repository security

- Access to code
- Workflow secrets
- Your code

Workflow secrets

@robbos81

Repository secrets

PUBLISH_PROFILE	Updated on Oct 26, 2019	Update	Remove
SONAR_TOKEN	Updated on Apr 11, 2020	Update	Remove

```
41
42      # publish to Azure App Service
43      - name: 'Run Azure webapp deploy action using publish profile credentials'
44        uses: azure/webapps-deploy@v2
45        with:
46          app-name: dotnetcorewebapp19 # Replace with your app
47          publish-profile: ${{ secrets.publish_profile }} # Define secret variable in repository settings as per action documentation
48          package: './dotnetcorewebapp'
49
```

Workflow secrets

Encrypted client side before reaching GitHub:

- Encrypted with the public key for your org or repo (created and stored by GitHub)
- Used when using the UI
- Encrypt yourself before posting to the REST API

Secrets are not shared to forked repositories

Who has access to your secrets?

For creating at repo level: Repository Owner access

For creating at org level: Admin access to the org

Set an access policy for the secrets:

- All repositories
- Private repositories
- Only selected repositories

Who has access to your secrets?

Encrypted until used, then injected as:

- An environment variable
- Direct input

Will be redacted in logs

Don’t use structured data (like json): hard to redact

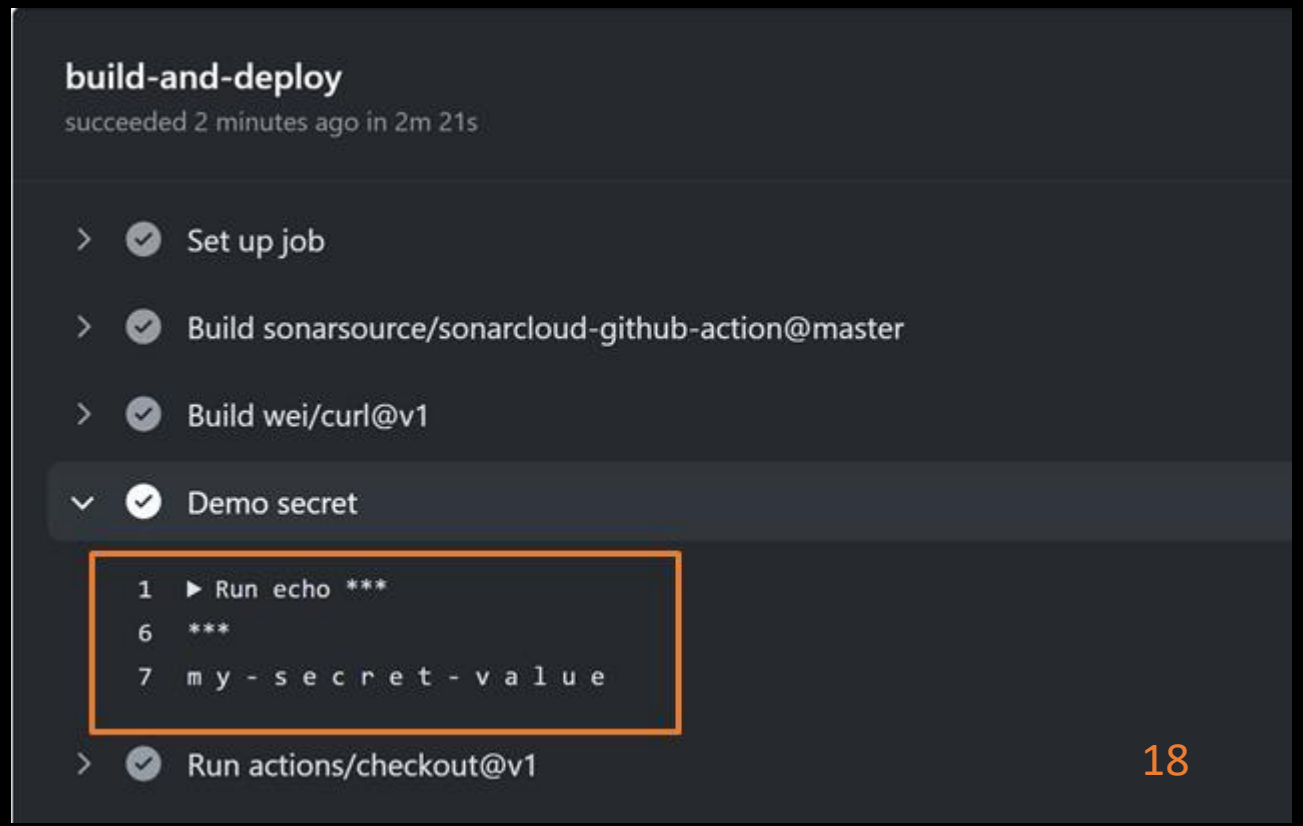
Who has access to your secrets?

- Actions can do anything with them!
- Anyone with access to the Action Logs should be considered to have access to your secrets

```yaml
jobs:
  build-and-deploy:

    runs-on: ubuntu-latest

    steps:
    - name: Demo secret
      run: |
        echo ${{ secrets.DEMO_LOG }}
        echo ${{ secrets.DEMO_LOG }} | sed 's/./& /g'
```



Repository security

- Access to code
- Workflow secrets
- Your code/repo

Your code

Anything in your repository:

- Workflow files
- Shell scripts
- Your own code
- Dependencies:
  - Packages
  - Containers

Best practices:

- Static code analysis
  - Check your own code!
- Third party dependency scanning
  - 99% of your code, is not yours:
    - Scan for known vulnerabilities
  - Keep your dependencies up to date!

Your code/repo – trace changes

Who made changes:

- Code: Git commit history

- Everything around your code is in the audit log

Your code/repo – trace changes (org level)

Audit log:

- Access
- Secrets
- Access Tokens
- OAuth grants
- Enabling features
- Etc.

GlobalDevOpsBootcamp

Repositories | Packages | People | Teams | Projects | Insights | Settings

GlobalDevOpsBo...
Organization settings

- Profile
- Billing & plans
- Member privileges
- Organization security
- Security & analysis
- Verified domains
- Audit log
- Webhooks
- Third-party access

Audit log

Filters | Search audit logs

Recent events

rajbos – team.add_member
Added themselves to the GlobalDevOpsBootcamp/demo-team team
Netherlands | 14 days ago

rajbos – team.create
Created the team GlobalDevOpsBootcamp/demo-team
Netherlands | 14 days ago

MOlausson – org_credential_authorization.grant
MOlausson authorized Personal Access Token ***** to access the
Sweden | on Dec 17, 2020

@robbos81

GitHub Actions Security

- Repository security
- Runners and security
- Actions and security

- Forking actions
- Keeping up to date

Workflow Runners

Actions execute on runners

Self hosted

- Cloud / On premises hosted by yourself
- OS + Tools update = YOUR responsibility
- Enables specific environment setup
- No usage limits

GitHub hosted

- OS + Tools update = GitHub's responsibility
- Per minute rating applies after the free minutes
- Clean execution environment with every run

```yaml
name: .NET Core Deploy to IIS

on:
  push:
    branches:
      - "self-hosted"

jobs:
  build-and-deploy:

    runs-on: self-hosted

    steps:
    - uses: actions/checkout@v1
    - name: Setup .NET Core
      uses: actions/setup-dotnet@v1
      with:
        dotnet-version: 3.0.100
```

```yaml
name: .NET Core

on: [push]

jobs:
  build-and-deploy:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v1
    - name: Setup .NET Core
      uses: actions/setup-dotnet@v1
      with:
        dotnet-version: 3.0.100
```

Workflow Runners

Security

- Environment scope
  - Network
  - Shared state between runs

- User: limit its access!

Best practice: Run the action inside of a container

```yaml
jobs:
  my_first_job:
    steps:
      - name: My first step
        uses: docker://gcr.io/cloud-builders/gradle
```

```yaml
jobs:
  test-box:
    runs-on: ubuntu-latest
    container:
      image: azul/zulu-openjdk-alpine:8-jre
    steps:
    - uses: actions/checkout@v2
    - name: What OS is running
      run: uname -a
    - name: What java version do we have
      run: java -version
```

Workflow runners

Best practice: Don’t use self hosted runners for public repositories

Example:

- Your repo
- New fork
- Adds malicious code
- Create pull request to your repo
- Workflow is executed on your self hosted runner?

Persisting data between runs

Run 1:

- Download dependencies
- Build the code
- Somehow overwrite the dependency cache

Run 2:

- Use cached dependencies
- Build the code
- Malicious dependency in build artefact

Solarwind attack:
https://xpir.it/Solorigate

Workflow runners – Best practice

Don't share runners (and machines!) between repositories:

- Run 1 can influence Run 2

Risks:

- Malicious programs
- Escaping the runner sandbox
- Exposing access to the (network) environment
- Persisting unwanted or dangerous data

GitHub Actions Security

Repository security

Runners and security

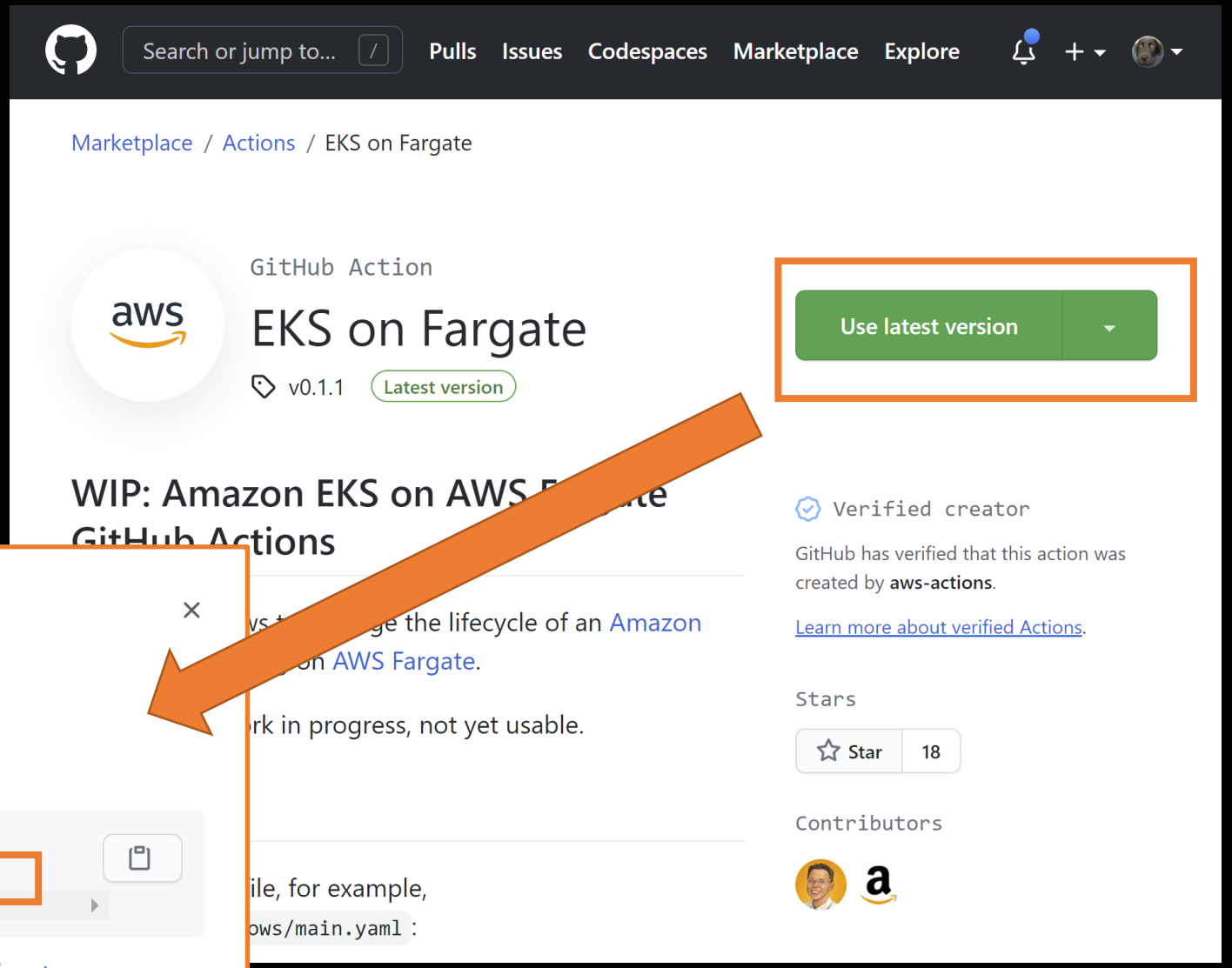
Actions and security

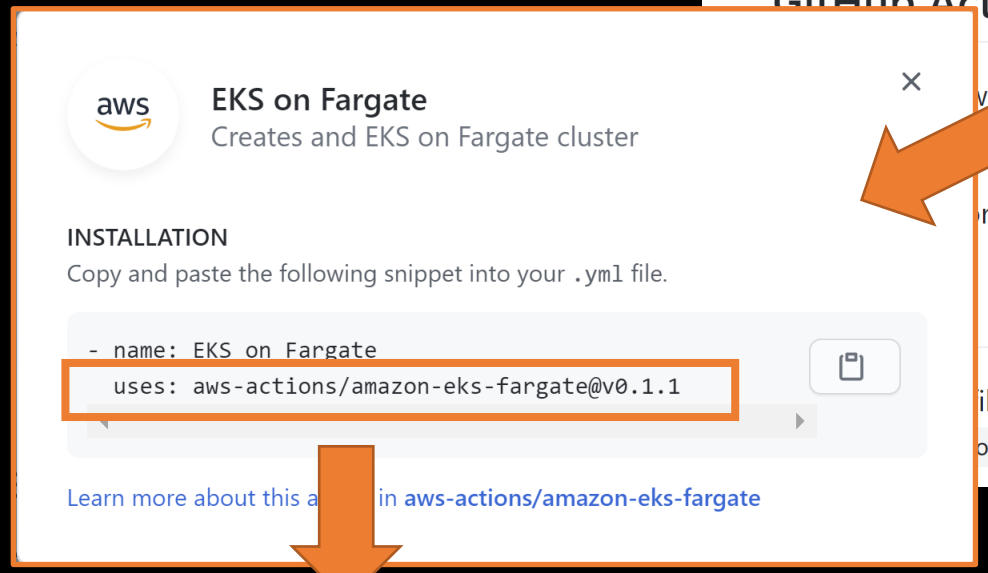
Forking actions

Keeping up to date

Actions

Marketplace or by direct url





https://github.com/aws-actions/amazon-eks-fargate

Actions and security

Are you running just any action from the internet?

SCARY, especially in an Enterprise or on local runners

Protective measures

```yaml
uses: shprink/nonharmful-and-must-have-actions@v1
with:
  my-secret: ${{ secrets.MY_SECRET }}
```

https://github.com/shprink/nonharmful-and-must-have-actions

If the repo has an action.yml, you can use it in your workflow

Protective measures

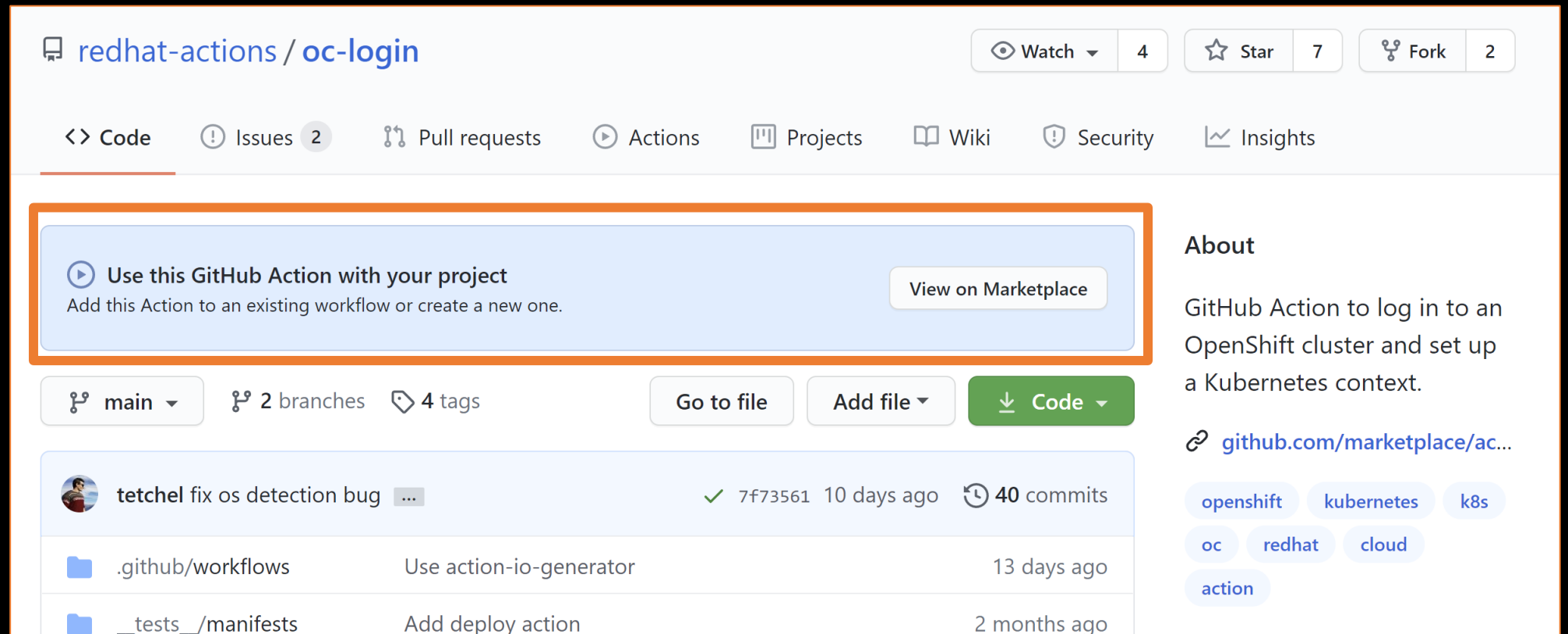
Manually:

1. Check the action repo code before use
2. Check its container images and dependencies before use

Protective measures

Only use actions listed in the marketplace?

- There is no real verification process for it ☹



Protective measures

Actions

An entirely new way to automate your development workflow.

45 results for "z" filtered by Actions ×

OWASP ZAP Baseline Scan
By zaproxy
Scans the web ap n with the OWASP ZAP Baseline Scan
135 stars

Zeebe Action
By jwulf
A GitHub action to interact with Zeebe and Camunda Cloud
6 stars

Verified creator

GitHub has verified that this action was created by **pachyderm**.

Learn more about verified Actions.

Protective measures

Limiting actions altogether

Actions permissions

- (●) **Allow all actions**
  Any action can be used, regardless of who authored it or whe
- ( ) **Disable Actions**
  The Actions tab is hidden and no workflows can run.
- ( ) **Allow local actions only**
  Only actions defined in a repository within rajbos can be used
- ( ) **Allow select actions**
  Only actions that match specified criteria can be used. Learn n

Actions permissions

- ( ) **Allow all actions**
  Any action can be used, regardless of who authored it or where it is defined.
- ( ) **Disable Actions**
  The Actions tab is hidden and no workflows can run.
- ( ) **Allow local actions only**
  Only actions defined in a repository within rajbos can be used.
- (●) **Allow select actions**
  Only actions that match specified criteria can be used. Learn more about allowing specific actions to run.

  - [x] Allow actions created by GitHub
  - [ ] Allow Marketplace actions by verified creators

  Allow specified actions

  rajbos-actions/*,

  Wildcards, tags, and SHAs are allowed. Examples: `monalisa/octocat@*`, `monalisa/octocat@v2`, `monalisa/*`

Protective measures

rajbos / dotnetcore-webapp

Unwatch 1 | Star 0 | Fork 110

Code | Issues | Pull requests | Actions | Projects | Wiki | Security 3 | Insights | ...

Updating actions with forks (#3) * Update dotnetcore.yml * Update dotnetcore.yml using actions from the rajbos-actions org .NET Core #94

Summary

Jobs

Triggered via push 18 seconds ago	Status	Total duration	Artifacts
rajbos pushed c64d658 main	Startup failure	–	–

Annotations

1 error

```
wei/curl@v1 is not allowed to be used in rajbos/dotnetcore-webapp.
Actions in this workflow must be: created by GitHub, within a
repository owned by rajbos or match the following: rajbos-actions/*.
```

.NET Core: .github#L1

Protective measures

Pin the action version:

```
uses: gaurav-nelson/github-action-markdown-link-check@v1
uses: gaurav-nelson/github-action-markdown-link-check@v1.0.1
```

Best practice: Pin the Action's commit SHA:

```
uses: gaurav-nelson/github-action-markdown-link-check@44a942b2f7ed0dc101d556f281e906fb79f1f478
```

Recommendation

- Best practice: Limit to local actions and fork action repositories
- Also create a separate org to test actions in, before forking them
  - To enable DevOps teams to have the autonomy to test and verify themselves

Workflow attack vectors

- Forks of public repos
- Common fields

Forks of public repos

```yaml
3   on:
4     - push
5     - pull_request
6     - pull_request_target
7
8   jobs:
9     build-and-deploy:
10      environment: PullRequestEnvironment
11
12      runs-on: ubuntu-latest
13
14      steps:
15      - uses: actions/checkout@v1
```

Safe, runs on merge commit, read only access

High risks! Runs on the target, has read + write access and can access secrets

https://xpir.it/gh-pwn-request

Common fields

```
github.event.issue.title
github.event.issue.body
github.event.pull_request.title
github.event.pull_request.body
github.event.comment.body
github.event.review.body
github.event.review_comment.body
github.event.pages.*.page_name
github.event.commits.*.message
github.event.head_commit.message
github.event.head_commit.author.email
github.event.head_commit.author.name
github.event.commits.*.author.email
github.event.commits.*.author.name
github.event.pull_request.head.ref
github.event.pull_request.head.label
github.event.pull_request.head.repo.default_branch
github.head_ref
```

Common fields

```
- name: Check title
  run: |
    title="${{ github.event.issue.title }}"
    if [[ ! $title =~ ^.*:\ .*$ ]]; then
      echo "Bad issue title"
      exit 1
    fi
```

Payload: `a"; echo test`

Remediation

```
- name: print title
  env:
    TITLE: ${{ github.event.issue.title }}
  run: echo "$TITLE"
```

https://xpir.it/actions-untrusted-input

GitHub Actions Security

Repository security

Runners and security

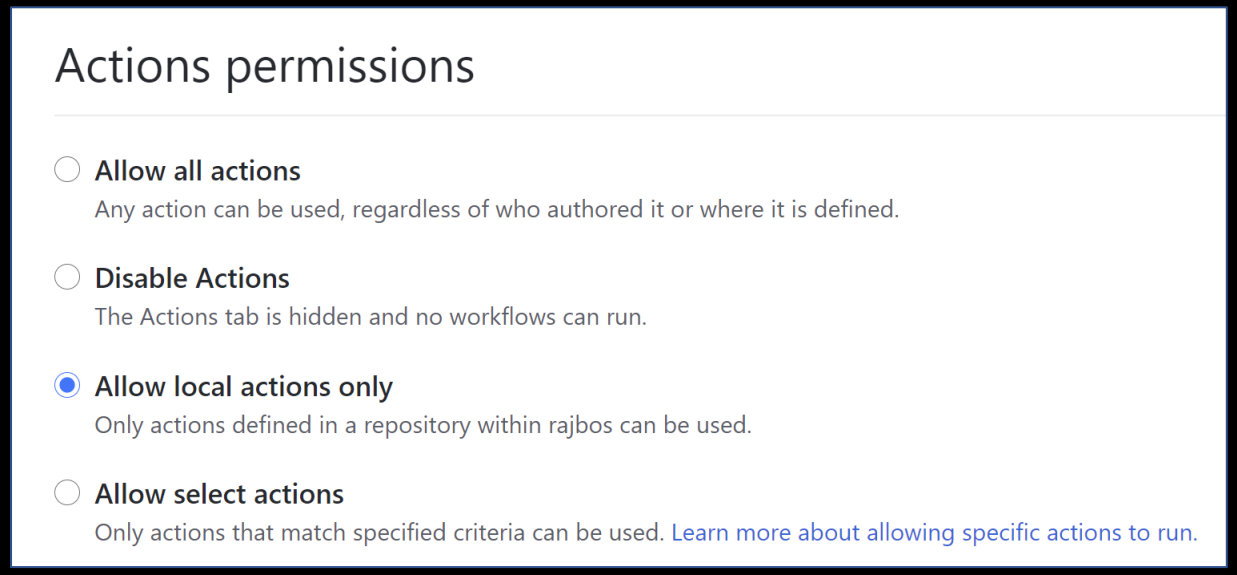
Actions and security

Forking actions

Keeping up to date

Forking actions

- Best practice: fork the action to a local (org) repo
- Limit actions to only local actions



Forking actions

Pros:

- More secure
- Backup of actions that can be deleted or moved to a different org/repo

Cons:

- More maintenance work
  - Fork needs to be created
  - Kept up to date
- Limits the usage of new actions in your org, as someone create the new action (and by that take responsibility for enabling its use)

GitHub Actions Security

Repository security

Runners and security

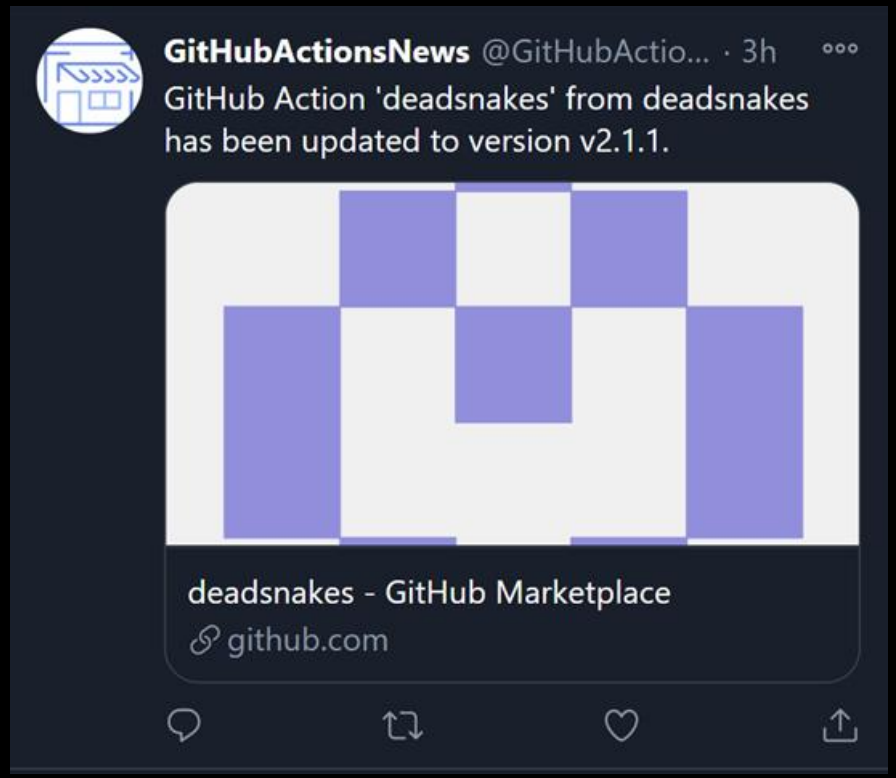
Actions and security

Forking actions

Keeping up to date

Staying up to date

Follow @githubactions on Twitter!



Update action versions

1. Review the Action
   Use Actions + Commit SHA + Dependabot

---

2. Review the Action
   Fork the Actions repo, update your forks and use Dependabot

Option 1: Use SHA + Dependabot

Best practice: Pin the Action's commit SHA:

uses: gaurav-nelson/github-action-markdown-link-check@44a942b2f7ed0dc101d556f281e906fb79f1f478

Add .github/dependabot.yml to the repo

```yaml
#Dependabot will check the dependencies in this repo for updates

version: 2
updates:
  - package-ecosystem: "github-actions"
    directory: "/"
    schedule:
      # Check for updates to GitHub Actions every weekday
      interval: "daily"


  - package-ecosystem: "nuget"
    directory: "/"
    schedule:
      # Check for updates to on nuget packages every weekday
      interval: "daily"
```

Use Dependabot

Search or jump to... / Pull requests Issues Marketplace Explore

rajbos / dotnetcore-webapp

Unwatch 1 Star 1 Fork 114

Code Issues Pull requests 5 Actions Projects Wiki Security 6 Insights

Bump rajbos-actions/trx-parser from v0.0.3 to v0.0.5 #5

Edit Open with

Open dependabot wants to merge 1 commit into main from dependabot/github_actions/rajbos-actions/trx-parser-v0.0.5

Conversation 1 Commits 1 Checks 3 Files changed 1 +1 −1

Changes from all commits File filter... Jump to... 0 / 1 files viewed Review changes

2 .github/workflows/dotnetcore.yml Viewed

```
@@ -78,7 +78,7 @@ jobs:
78 78
79 79        # Using the trx-parser action
80 80        - name: Parse Trx files
81    -        uses: rajbos-actions/trx-parser@v0.0.3
   81 +        uses: rajbos-actions/trx-parser@v0.0.5
82 82          id: trx-parser
83 83          with:
84 84            TRX_PATH: ${{ github.workspace }}\\dotnet-core-webapp.webtests\TestResults #This should be the path to your TRX files
```

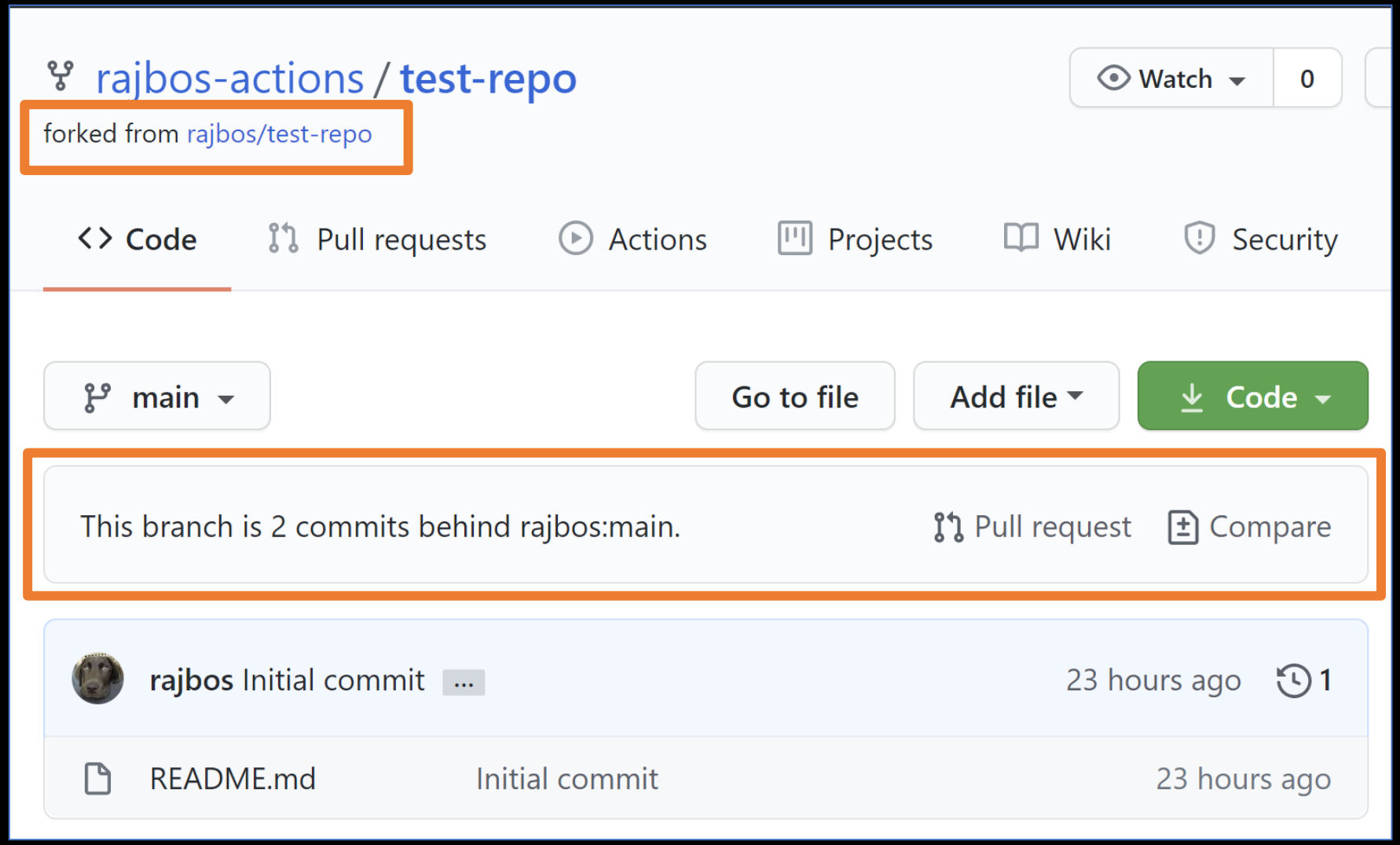
Update action versions

1. Review the Action
   Use Actions + Commit SHA + Dependabot

---

2. Review the Action
   Fork the Actions repo, update your forks and use Dependabot

Keep you forked action up to date



Keep your forked action up to date

Fork a repo and automate it!

https://github.com/rajbos/github-fork-updater

Contains:

- Scheduled workflow
- Creates an issue
- Review the changes
- Label the issue
- Pull in changes

Creates issues

Search or jump to... / Pulls Issues Codespaces Marketplace Explore

rajbos / github-fork-updater

Unwatch 1 Star 0 Fork 0

Code Issues 7 Pull requests Actions Projects Wiki ...

Parent repository for [rajbos/SonarQube-AzureAppService] has updates available #25

Edit New issue

Open github-actions bot opened this issue 22 hours ago · 0 comments

github-actions bot commented 22 hours ago

The parent repository for **rajbos/SonarQube-AzureAppService** has updates available.

Important!

Click on this compare link to check the incoming changes before updating the fork.

To update the fork

Add the label **update-fork** to this issue to update the fork

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Review before merging

rajbos / SonarQube-AzureAppService

forked from vanderby/SonarQube-AzureAppService

Unwatch 1 | Star 0 | Fork 65

Code | Pull requests | Actions | Projects | Security | Insights | ...

This is a direct comparison between two commits made in this repository or its related repositories. View the default comparison for this range here.

Comparing changes

base repository: rajbos/SonarQube-AzureAppS... | base: master

head repository: vanderby/SonarQube-AzureAp... | compare: master

Showing 5 changed files with 283 additions and 44 deletions. Unified Split

8 .gitignore

```
@@ -1,6 +1,9 @@
## Ignore Visual Studio temporary files, build results, and
## files generated by popular Visual Studio add-ons.

+ # Don't include extracted sonarqube folder
+ sonarqube-*/
```

Automation

- Add a label
- Fork gets updated
- Issue gets closed

Parent repository for [rajbos/ParallelTestRunner] has updates available #23

Closed github-actions bot opened this issue 2 days ago · 2 comments

github-actions bot commented 2 days ago

The parent repository for **rajbos/ParallelTestRunner** has updates available.

Important!

Click on this compare link to check the incoming changes before updating the fork.

To update the fork

Add the label **update-fork** to this issue to update the fork

rajbos added the update-fork label now

rajbos commented now

Updating the fork with the incoming changes from the parent repository

rajbos commented now

Fork has been updated

rajbos closed this now

Pros of forking

- Backup of the action
- Full control over updates
- Pull in updates with validation centrally
- Only allow actions from your actions organization

- Skip commit SHA lookup and updating in every workflow
- Skip adding Dependabot in every repository

GitHub Actions Security

Repository security
Runners and security
Actions and security

Forking actions
Keeping up to date

Best practices summarized

- Treat workflow secrets very carefully: best to think of them as public
- Review actions' source code
- Pin actions to commit SHA
- Don't trust incoming Pull Requests on public repos
- Fork the action repo and limit actions to local actions only
- Have an organization setup to test with
- Keep your forked actions up to date

Thank you!

SOLIDIFY

Rob Bos
DevOps Consultant – Xpirit
The Netherlands

Next Solidify show

- 2021-06-04, #4, Modernizing your applications with Microsoft Azure

SOLIDIFY

www.solidify.se

Contact us at info@solidify.se if you want to know more or need help