

# How to use GitHub Actions with security in mind



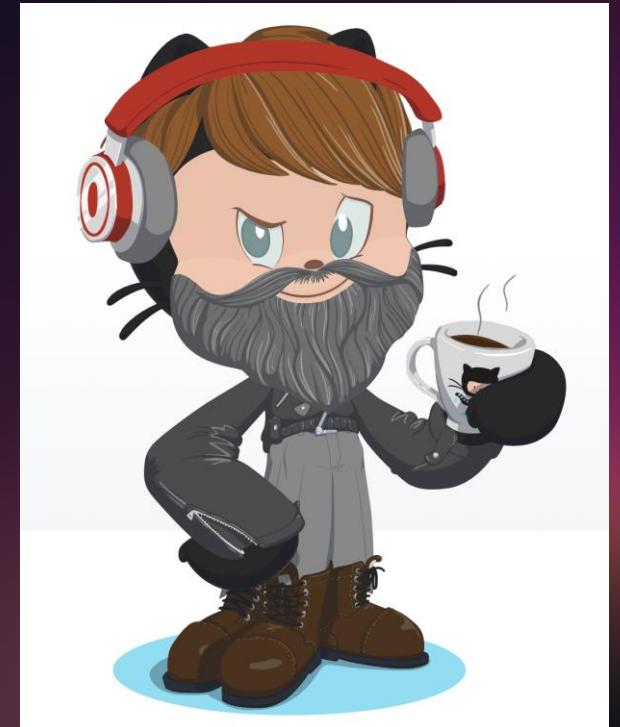
@robbos81  
<https://devopsjournal.io>

<https://myoctocat.com>

# How to use GitHub Actions with security in mind

Rob Bos  
DevOps Consultant – Xpirit  
The Netherlands

@robbos81  
<https://devopsjournal.io>



<https://myoctocat.com>

# Words matter

## Workflows:

Execute one or more **Actions**

Workflows triggered by events:

- Push
- Creating an issue
- Release

Execute on a runner

## Actions:

Steps in the **workflows**

Basis: Run a shell script

Create your own

Use an existing one from the marketplace



## Repository security

Runners and security

Actions and security

Forking actions

Keeping up to date

@robbos81

# Repository security

Access to code

Workflow secrets

Your code

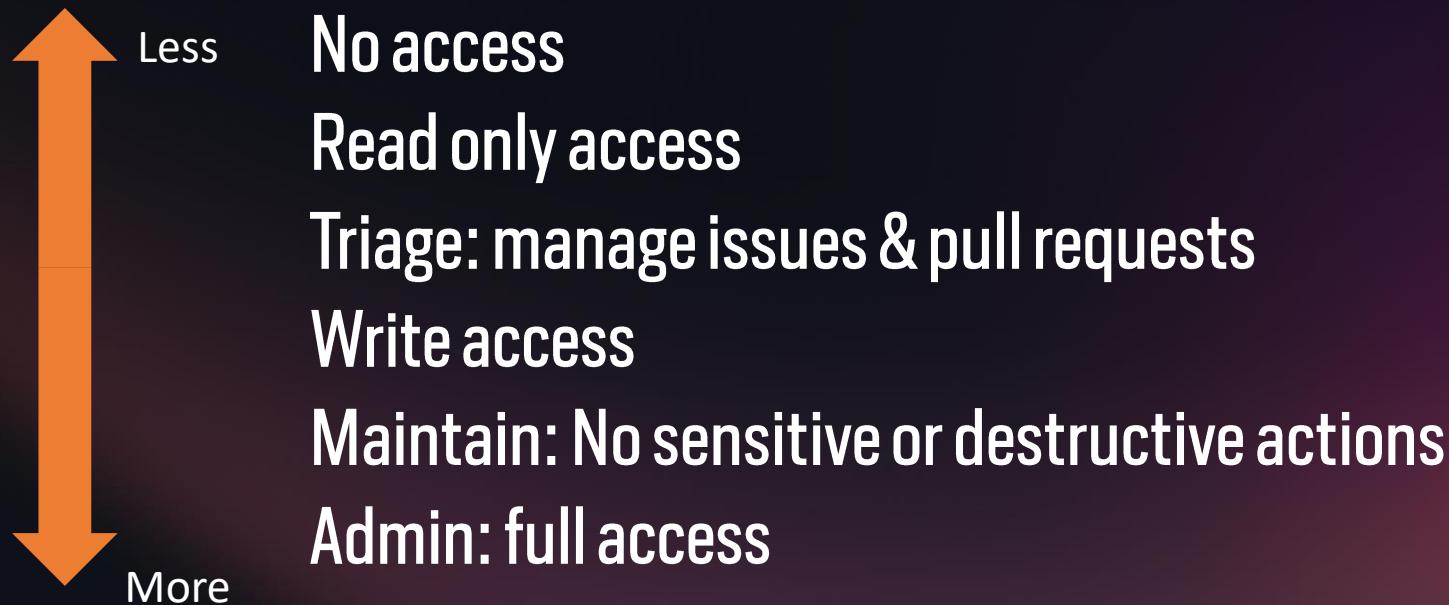
# Code – Who has access?

Access levels can be set at:

- Repository
- Organization
- Enterprise

# Code – Who has access?

## Permission levels



# 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.

@robbos81

The screenshot shows the GitHub organization settings page for 'GlobalDevOpsBootcamp'. The left sidebar lists various organization settings like Profile, Billing & plans, Member privileges, etc., with 'Audit log' highlighted by an orange box. The main content area shows the 'Audit log' section with a search bar and a list of recent events. One event is highlighted with an orange box: 'rajbos – team.add\_member' where rajbos added themselves to the 'GlobalDevOpsBootcamp/demo-team' team in the Netherlands 14 days ago. Another event listed is 'rajbos – team.create' where rajbos created the team 'GlobalDevOpsBootcamp/demo-team' in the Netherlands 14 days ago. The bottom right corner of the slide has the number '16'.

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

16

# Repository security

Access to code

Workflow secrets

Your code

# Workflow secrets

@robbos81

## Repository secrets

 PUBLISH\_PROFILE

Updated on Oct 26, 2019

Update

Remove

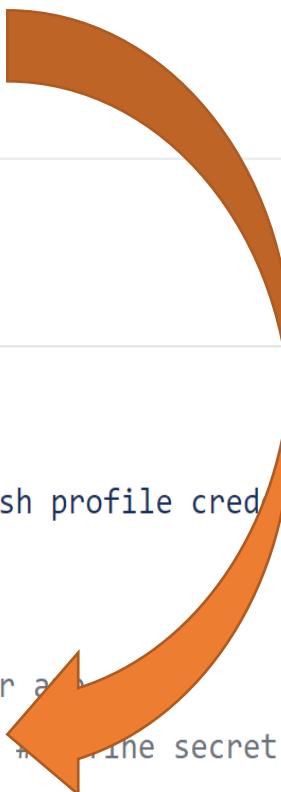
 SONAR\_TOKEN

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 name  
47          publish-profile: ${{ secrets.publish_profile }} # Define the secret variable in repository settings as per action documentation  
48          package: './dotnetcorewebapp'
```



# 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

```
5 jobs:  
6   build-and-deploy:  
7  
8     runs-on: ubuntu-latest  
9  
10    steps:  
11      - name: Demo secret  
12        run: |  
13          echo ${{ secrets.DEMO_LOG }}  
14          echo ${{ secrets.DEMO_LOG }} | sed 's/./& /g'
```



build-and-deploy  
succeeded 2 minutes ago in 2m 21s

- >  Set up job
- >  Build sonarsource/sonarcloud-github-action@master
- >  Build wei/curl@v1
- >  Demo secret
  - 1 ► Run echo \*\*\*
  - 6 \*\*\*
  - 7 my - s e c r e t - v a l u e
- >  Run actions/checkout@v1

# 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!

**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

```
1 name: .NET Core Deploy to IIS
2
3 on:
4   push:
5     branches:
6       - "self-hosted"
7
8 jobs:
9   build-and-deploy:
10
11   runs-on: self-hosted
12
13 steps:
14   - uses: actions/checkout@v1
15   - name: Setup .NET Core
16     uses: actions/setup-dotnet@v1
17     with:
18       dotnet-version: 3.0.100
19
```

```
1 name: .NET Core
2
3 on: [push]
4
5 jobs:
6   build-and-deploy:
7
8   runs-on: ubuntu-latest
9
10 steps:
11   - uses: actions/checkout@v1
12   - name: Setup .NET Core
13     uses: actions/setup-dotnet@v1
14     with:
15       dotnet-version: 3.0.100
16
```

# Workflow Runners

## Security

- Environment scope
  - Network
  - Shared state between runs
- User: limit its access!

# Best practice: Run the action inside of a container

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



```
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

Solarwind attack:

<https://xpir.it/solorigate>

Run 2:

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

# 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

<https://xpir.it/actions-kubernetes>

Repository security  
Runners and security  
**Actions and security**

Forking actions  
Keeping up to date

# Actions

Marketplace or by direct url

EKS on Fargate  
Creates and EKS on Fargate cluster

INSTALLATION

Copy and paste the following snippet into your .yml file.

```
- name: EKS on Fargate
  uses: aws-actions/amazon-eks-fargate@v0.1.1
```

Learn more about this action in [aws-actions/amazon-eks-fargate](#)

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

Marketplace / Actions / EKS on Fargate

GitHub Action

**EKS on Fargate**

v0.1.1 Latest version

WIP: Amazon EKS on AWS Fargate GitHub Actions

This action manages the lifecycle of an Amazon EKS cluster on AWS Fargate.

This action is work in progress, not yet usable.

Verified creator

GitHub has verified that this action was created by **aws-actions**.

[Learn more about verified Actions.](#)

Stars

Star 18

Contributors

# Actions and security



Are you running just any  
action from the internet?



Scary! Especially in an  
enterprise or on local runners

# 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 😞

The screenshot shows a GitHub repository page for 'redhat-actions / oc-login'. The top navigation bar includes links for Code, Issues (2), Pull requests, Actions, Projects, Wiki, Security, and Insights. The 'Actions' tab is currently selected. A prominent call-to-action box is highlighted with an orange border, containing the text 'Use this GitHub Action with your project' and a 'View on Marketplace' button. Below this, the repository details show 'main' branch, 2 branches, 4 tags, and a commit history with three recent commits. To the right, there's an 'About' section describing the action as a GitHub Action to log in to an OpenShift cluster and set up a Kubernetes context, along with a link to the marketplace listing and relevant tags like openshift, kubernetes, k8s, oc, redhat, cloud, and action.

Code Issues 2 Pull requests Actions Projects Wiki Security Insights

Watch 4 Star 7 Fork 2

Use this GitHub Action with your project  
Add this Action to an existing workflow or create a new one.  
View on Marketplace

main 2 branches 4 tags Go to file Add file Code

tetchel fix os detection bug ... ✓ 7f73561 10 days ago 40 commits

.github/workflows Use action-io-generator 13 days ago

\_tests\_/manifests Add deploy action 2 months ago

About

GitHub Action to log in to an OpenShift cluster and set up a Kubernetes context.

[github.com/marketplace/actions/redhat-actions/oc-login](https://github.com/marketplace/actions/redhat-actions/oc-login)

openshift kubernetes k8s

oc redhat cloud

action

@robbos81

# Protective measures

## Actions

An entirely new way to automate your development workflow.

45 results for "z" filtered by [Actions](#) [x](#)



[OWASP ZAP Baseline Scan](#)

By zaproxy

Scans the web application with the OWASP ZAP Baseline Scan

135 stars



[Zeebe Action](#)

By jwulf

A GitHub action to interact with Zeebe and Camunda Cloud

6 stars

ed creators

[Verified creator](#)

GitHub has verified that this action was created by [pachyderm](#).

[Learn more about verified Actions.](#)

# Verified Creator

Verification process:

- GitHub Profile information is present and accurate
- Two factor authentication is on for the organization
- Domain verification through a txt record

See: <https://xpir.it/verified-publisher>

# Protective measures

Limiting actions altogether

## 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.](#)

## 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.](#)

#### 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

The screenshot shows a GitHub repository page for `rajbos / dotnetcore-webapp`. The `Actions` tab is selected. A recent run is displayed:

- Status:** Startup failure
- Triggered via push 18 seconds ago**
- Pushed by:** `rajbos` (commit `c64d658` on `main`)
- Total duration:** -
- Artifacts:** -

**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/*`.

# 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:

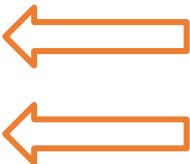
```
uses: gaurav-nelson/github-action-markdown-link-check@44a942b2f7ed0dc101d556f281e906fb79f1f478
```

# Workflow attack vectors

- Forks of public repos
- Common fields

# Forks of public repos

```
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>

# Pull Requests

`${{ secrets.GITHUB_TOKEN }}`

## Workflow permissions

Choose the default permissions granted to the GITHUB\_TOKEN when running workflows in this repository. You can specify more granular permissions in the workflow using YAML. [Learn more](#).

**Read and write permissions**

Workflows have read and write permissions in the repository for all scopes.

**Read repository contents permission**

Workflows have read permissions in the repository for the contents scope only.

# Pull Requests

```
${{ secrets.GITHUB_TOKEN }}
```

```
name: Pull request labeler

on: [ pull_request_target ]

permissions:
  contents: read
  pull-requests: write

jobs:
  triage:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/labeled@v2
        with:
          repo-token: ${{ secrets.GITHUB_TOKEN }}
```

# 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>

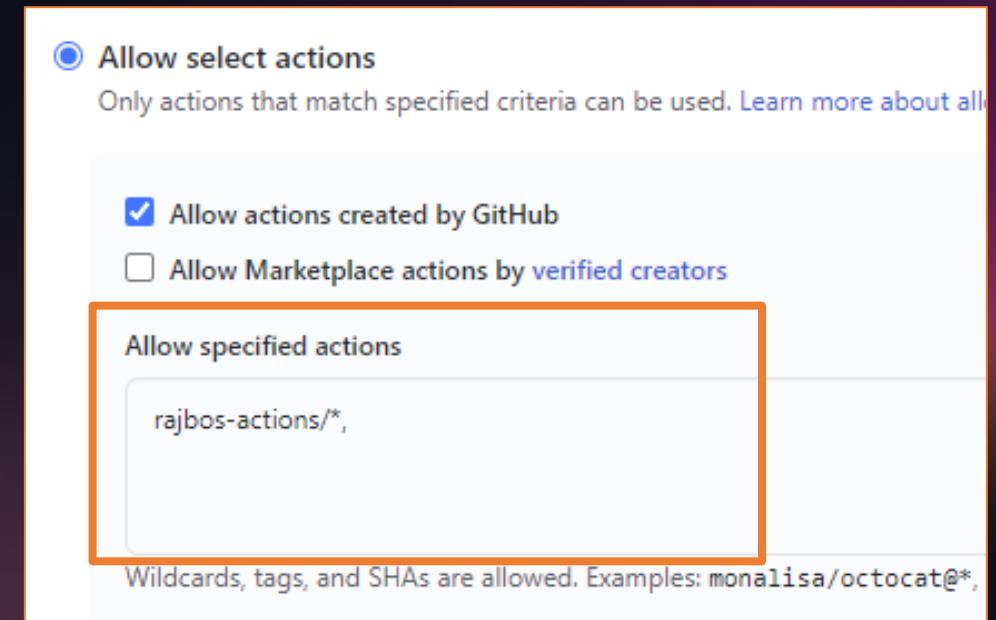
Repository security  
Runners and security  
Actions and security

Forking actions  
Keeping up to date

# Forking actions

Best practice: fork the action to a local organization

Limit actions to only select actions from that organization



# 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

Repository security  
Runners and security  
Actions and security

Forking actions  
Keeping up to date

# 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

```
1 #Dependabot will check the dependencies in this repo for updates
2
3 version: 2
4 updates:
5   - package-ecosystem: "github-actions"
6     directory: "/"
7     schedule:
8       - # Check for updates to GitHub Actions every weekday
9         interval: "daily"
10
11
12   - package-ecosystem: "nuget"
13     directory: "/"
14     schedule:
15       - # Check for updates to on nuget packages every weekday
16         interval: "daily"
```



# Use Dependabot

The screenshot shows a GitHub repository page for `rajbos / dotnetcore-webapp`. The `Pull requests` tab is selected, displaying a single pull request titled `Bump rajbos-actions/trx-parser from v0.0.3 to v0.0.5 #5`. The pull request has been merged into the `main` branch from the `dependabot/github_actions/rajbos-actions/trx-parser-v0.0.5` branch. The changes are shown in the `.github/workflows/dotnetcore.yml` file, specifically in the `jobs` section. The code diff highlights the change from `uses: rajbos-actions/trx-parser@v0.0.3` to `uses: rajbos-actions/trx-parser@v0.0.5`.

```
diff --git a/.github/workflows/dotnetcore.yml b/.github/workflows/dotnetcore.yml
--- a/.github/workflows/dotnetcore.yml
+++ b/.github/workflows/dotnetcore.yml
@@ -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 your forked action up to date

The screenshot shows a GitHub repository page for `rajbos-actions / test-repo`. The repository is a fork of `rajbos/test-repo`. The main navigation bar includes links for Code, Pull requests, Actions, Projects, Wiki, and Security. Below the navigation, there are buttons for main branch selection, Go to file, Add file, and a green Code dropdown. A prominent message states: "This branch is 2 commits behind rajbos:main." It includes a Pull request button and a Compare link. The commit history shows two recent commits: "rajbos Initial commit" and "README.md Initial commit", both made 23 hours ago.

forked from [rajbos/test-repo](#)

<> Code Pull requests Actions Projects Wiki Security

main ▾ Go to file Add file ▾ Code ▾

This branch is 2 commits behind rajbos:main. Pull request Compare

rajbos Initial commit ... 23 hours ago 1

README.md Initial commit 23 hours ago

# 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

The screenshot shows a GitHub repository page for `rajbos / github-fork-updater`. The `Issues` tab is selected, showing 7 issues. The main content area displays an issue titled "Parent repository for [rajbos/SonarQube-AzureAppService] has updates available #25". The issue was opened by `github-actions (bot)` 22 hours ago with 0 comments. The issue body contains a message from `github-actions (bot)` stating: "The parent repository for `rajbos/SonarQube-AzureAppService` has updates available." Below this, there is an "Important!" section with a callout: "Click on this [compare link](#) to check the incoming changes before updating the fork." This callout is highlighted with an orange border. To the right of the issue body, there are sections for "Assignees" (None yet), "Labels" (None yet), "Projects" (None yet), and "Milestone" (None yet). At the bottom of the issue view, there is a note: "Add the label `update-fork` to this issue to update the fork".

# Review before merging

The screenshot shows a GitHub repository page for `rajbos/SonarQube-AzureAppService`. The repository was forked from `vanderby/SonarQube-AzureAppService`. The main navigation bar includes links for Code, Pull requests, Actions, Projects, Security, Insights, and more.

A message at the top states: "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

The comparison settings are highlighted with an orange box:

- base repository: `rajbos/SonarQube-AzureAppS...`
- base: `master`
- head repository: `vanderby/SonarQube-AzureAp...`
- compare: `master`

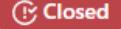
Below the comparison controls, it says "Showing 5 changed files with 283 additions and 44 deletions." and provides "Unified" and "Split" view options.

The diff view shows a file named `.gitignore` with 8 changes. The changes are as follows:

```
@@ -1,6 +1,9 @@
 1   1      ## Ignore Visual Studio temporary files, build results, and
 2   2      ## files generated by popular Visual Studio add-ons.
 3   3
 4   4      # Don't include extracted sonarqube folder
 5   5      +sonarqube-*/
```

1. Add a label
2. Fork gets updated
3. 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

# How to use GitHub Actions with security in mind

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

<https://xpir.it/actions-best-practices>

# Thank you!

Rob Bos  
DevOps Consultant - Xpirit  
The Netherlands

@robbos81  
<https://devopsjournal.io>



<https://myoctocat.com>