

# 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://twitter.com/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

# Repository security

Access to code

Workflow secrets

Your code

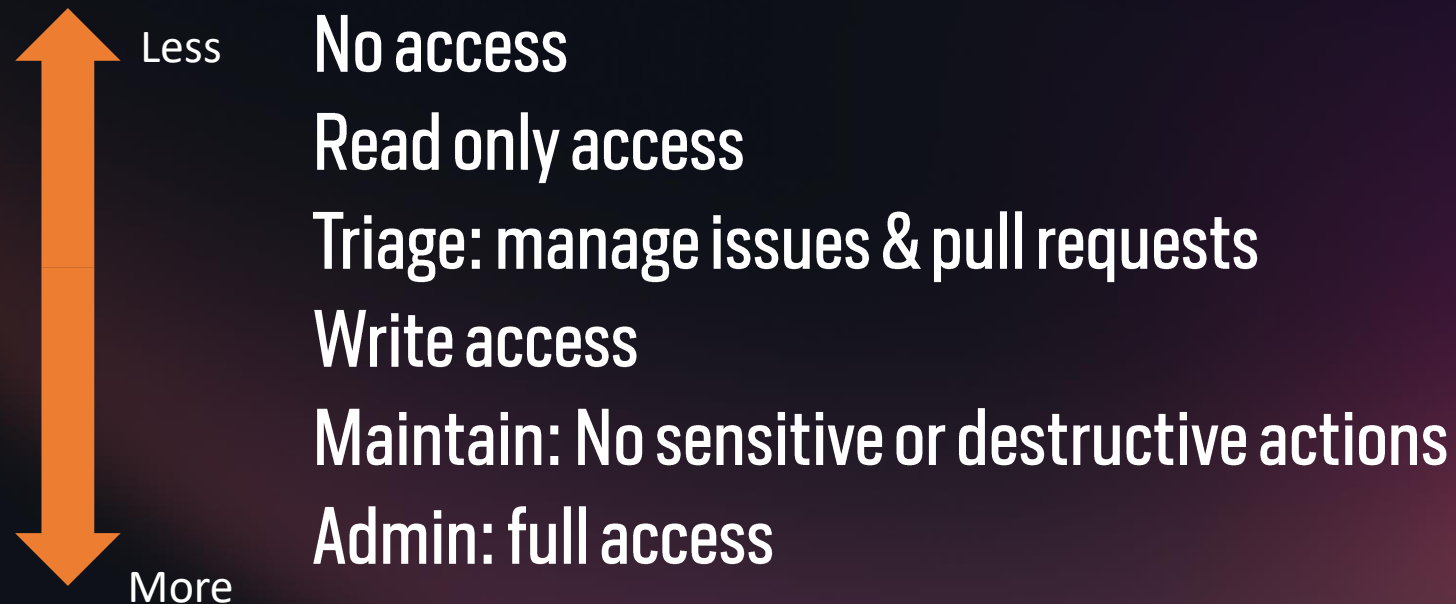
# Code – Who has access?

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.

The screenshot shows the GitHub organization settings page for 'GlobalDevOpsBootcamp'. The 'Settings' tab is highlighted with an orange box. In the left-hand navigation menu, the 'Audit log' option is also highlighted with an orange box. The main content area displays the 'Audit log' section, which includes a search bar and a list of recent events. The events listed are:

- rajbos – team.add\_member**: Added themselves to the `GlobalDevOpsBootcamp/demo-team` team. Location: Netherlands. Time: 14 days ago.
- rajbos – team.create**: Created the team `GlobalDevOpsBootcamp/demo-team`. Location: Netherlands. Time: 14 days ago.
- MOlausson – org\_credential\_authorization.grant**: MOlausson authorized Personal Access Token \*\*\*\*\* to access the organization. Location: Sweden. Time: on Dec 17, 2020.

# Repository security

Access to code

Workflow secrets

Your code

# Workflow secrets

@robbos81

## Repository secrets

 PUBLISH_PROFILE	Updated on Oct 26, 2019	<a href="#">Update</a>	<a href="#">Remove</a>
 SONAR_TOKEN	Updated on Apr 11, 2020	<a href="#">Update</a>	<a href="#">Remove</a>

```
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 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

```
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'
15
```



## 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 m y - 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

## 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

<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

The screenshot shows the GitHub Marketplace page for the 'EKS on Fargate' action. At the top, there is a search bar and navigation links for Pulls, Issues, Codespaces, Marketplace, and Explore. The main heading is 'Marketplace / Actions / EKS on Fargate'. Below this, the action is identified as a 'GitHub Action' by 'aws-actions' with the title 'EKS on Fargate'. A green button labeled 'Use latest version' is highlighted with an orange box. Below the button, the version 'v0.1.1' is shown with a 'Latest version' badge. The description reads 'WIP: Amazon EKS on AWS Fargate' and 'GitHub Actions'. A note states 'Work in progress, not yet usable.' The 'Verified creator' section indicates that GitHub has verified the action was created by 'aws-actions'. The 'Stars' section shows 18 stars. The 'Contributors' section lists the creator 'aws-actions'.

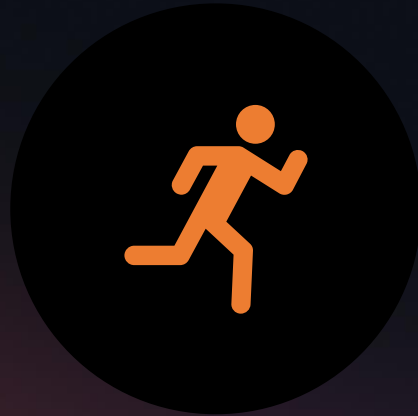
This screenshot shows the installation instructions for the 'EKS on Fargate' action. It includes the AWS logo and the title 'EKS on Fargate' with the description 'Creates and EKS on Fargate cluster'. Under the 'INSTALLATION' section, it instructs to 'Copy and paste the following snippet into your .yaml file.' A code block is shown with the following content: 

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

 The 'uses' line is highlighted with an orange box. Below the code block, there is a link to 'Learn more about this action in aws-actions/amazon-eks-fargate'.

<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

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 the GitHub repository page for 'redhat-actions / oc-login'. At the top, there are navigation links for 'Code', 'Issues' (2), 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', and 'Insights'. On the right, there are buttons for 'Watch' (4), 'Star' (7), and 'Fork' (2). A prominent blue box with an orange border highlights the 'Use this GitHub Action with your project' section, which includes the text 'Add this Action to an existing workflow or create a new one.' and a 'View on Marketplace' button. Below this, there are repository details: 'main' branch, '2 branches', and '4 tags'. A list of recent commits is visible, including one by 'tetchel' titled 'fix os detection bug' with 40 commits. The right sidebar shows an 'About' section with a description: 'GitHub Action to log in to an OpenShift cluster and set up a Kubernetes context.' and a link to the marketplace. There are also tags for 'openshift', 'kubernetes', 'k8s', 'oc', 'redhat', 'cloud', and 'action'.

# 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



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](#)

### 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 Actions workflow run that has failed. The workflow is titled "Updating actions with forks (#3) \* Update dotnetcore.yml \* Update dotnetcore.yml using actions from the rajbos-actions org .NET Core #94". The workflow was triggered via a push 18 seconds ago by user 'rajbos' on the 'main' branch. The status is "Startup failure".

The error message, highlighted with an orange box, states: "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/\*." The error is associated with the file ".NET Core: .github#L1".

Navigation links at the top include: Code, Issues, Pull requests, Actions (selected), Projects, Wiki, Security (3), and Insights.

Repository statistics: Unwatch (1), Star (0), Fork (110).

# 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/labeler@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>

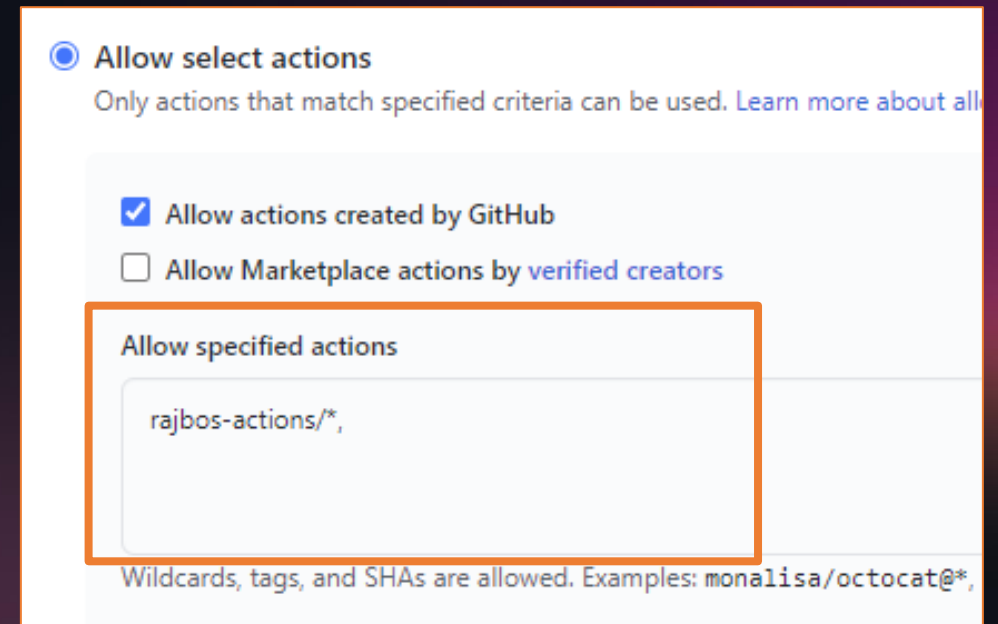
Repository security  
Runners and security  
Actions and security

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



Allow select actions  
Only actions that match specified criteria can be used. [Learn more about all](#)

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@\*,

# 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 pull request for the repository `rajbos / dotnetcore-webapp`. The pull request title is `Bump rajbos-actions/trx-parser from v0.0.3 to v0.0.5 #5`. The pull request is created by `dependabot` and is intended to merge into the `main` branch. The diff shows a change in the `.github/workflows/dotnetcore.yml` file, where the `uses` property of the `trx-parser` action is updated from `v0.0.3` to `v0.0.5`.

```
@@ -78,7 +78,7 @@ jobs:
 78 78
 79 79     # Using the trx-parser action
 80 80     - name: Parse Trx files
 81 81     - uses: rajbos-actions/trx-parser@v0.0.3
 81 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 the GitHub interface for a repository named 'rajbos-actions / test-repo'. The repository is a fork of 'rajbos/test-repo', as indicated by the text 'forked from rajbos/test-repo' which is highlighted with an orange box. The repository is currently on the 'main' branch. The interface includes navigation tabs for 'Code', 'Pull requests', 'Actions', 'Projects', 'Wiki', and 'Security'. Below these are buttons for 'main', 'Go to file', 'Add file', and 'Code'. A prominent orange box highlights a status message: 'This branch is 2 commits behind rajbos:main.' To the right of this message are links for 'Pull request' and 'Compare'. Below the status message, a commit history is shown, starting with 'rajbos Initial commit' from 23 hours ago, which includes a file named 'README.md'.

# 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. A specific issue is highlighted with the title 'Parent repository for [rajbos/SonarQube-AzureAppService] has updates available #25'. The issue was opened by 'github-actions' 22 hours ago. A comment from 'github-actions' is visible, stating: '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'. The 'compare link' is highlighted with an orange box. On the right side, there are settings for Assignees, Labels, Projects, and Milestone.

# Review before merging

The screenshot shows a GitHub pull request interface. At the top, the repository name is 'rajbos / SonarQube-AzureAppService', which is a fork of 'vanderby/SonarQube-AzureAppService'. The interface includes navigation tabs for Code, Pull requests, Actions, Projects, Security, and Insights. A message 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](#).' Below this is the 'Comparing changes' section, which contains a comparison selector. This selector is highlighted with an orange box and shows: 'base repository: rajbos/SonarQube-AzureAppS...' with 'base: master' selected, and 'head repository: vanderby/SonarQube-AzureAp...' with 'compare: master' selected. Below the selector, it says 'Showing 5 changed files with 283 additions and 44 deletions.' The file list shows a diff for '.gitignore', with lines 4 and 5 highlighted in green, indicating additions: '+ # Don't include extracted sonarqube folder' and '+ sonarqube-\*/'.

# 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

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

# 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



<https://myoctocat.com>

@robbos81

<https://devopsjournal.io>