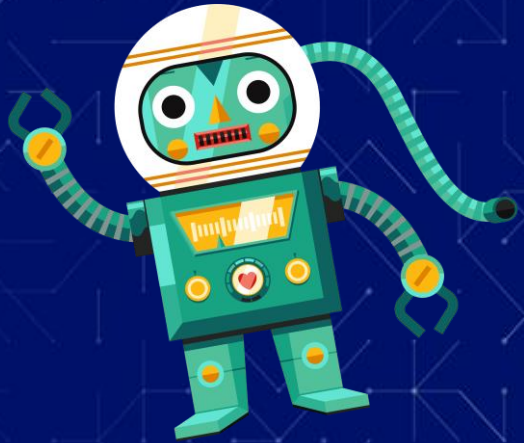


GitHub Actions Security

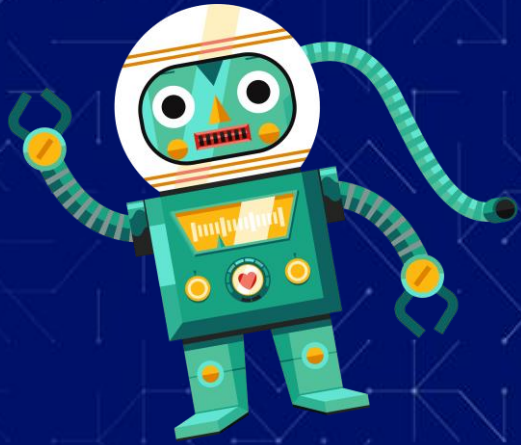


TECHORAMA



<https://myoctocat.com>

GitHub Actions Security



TECHORAMA

Rob Bos

DevOps Consultant – Xpirit

The Netherlands

devopsjournal.io

@robbos81



<https://myoctocat.com>

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



Search or jump to...



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



[Marketplace](#) / Search results

Types

Search for apps and actions

Apps

Actions



Categories

API management

Chat

Code quality

Code review

Continuous integration

Dependency management

Deployment

Actions

An entirely new way to automate your development workflow.

8436 results filtered by Actions



Deploy to Cloud Run

By google-github-actions

Use this action to deploy a container in the Google Container Registry to Cloud Run
53 stars



Buildah Build

By redhat-actions

Build a container image, with or without a Dockerfile
36 stars



Amazon ECS "Deploy Task Definition" Action for GitHub Actions

By aws-actions

Registers an Amazon ECS task definition, and deploys it to an ECS service
228 stars



Glo Add Label To Cards

By Axosoft

GitHub action to add a label to Glo Boards cards
3 stars

Workflow example

main [dotnetcore-webapp / .github / workflows / dotnetcore.yml](#)

```
1  name: .NET Core
2
3  on: [push]
4
5  jobs:
6    build-and-deploy:
7      environment: Production
8
9      runs-on: ubuntu-latest
10
11     steps:
12       - uses: actions/checkout@v1
13
14       - name: Setup .NET Core
15         uses: actions/setup-dotnet@v1
16         with:
17           dotnet-version: 3.0.100
18
19       # dotnet build
20       - name: Build with dotnet
21         run: |
22           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

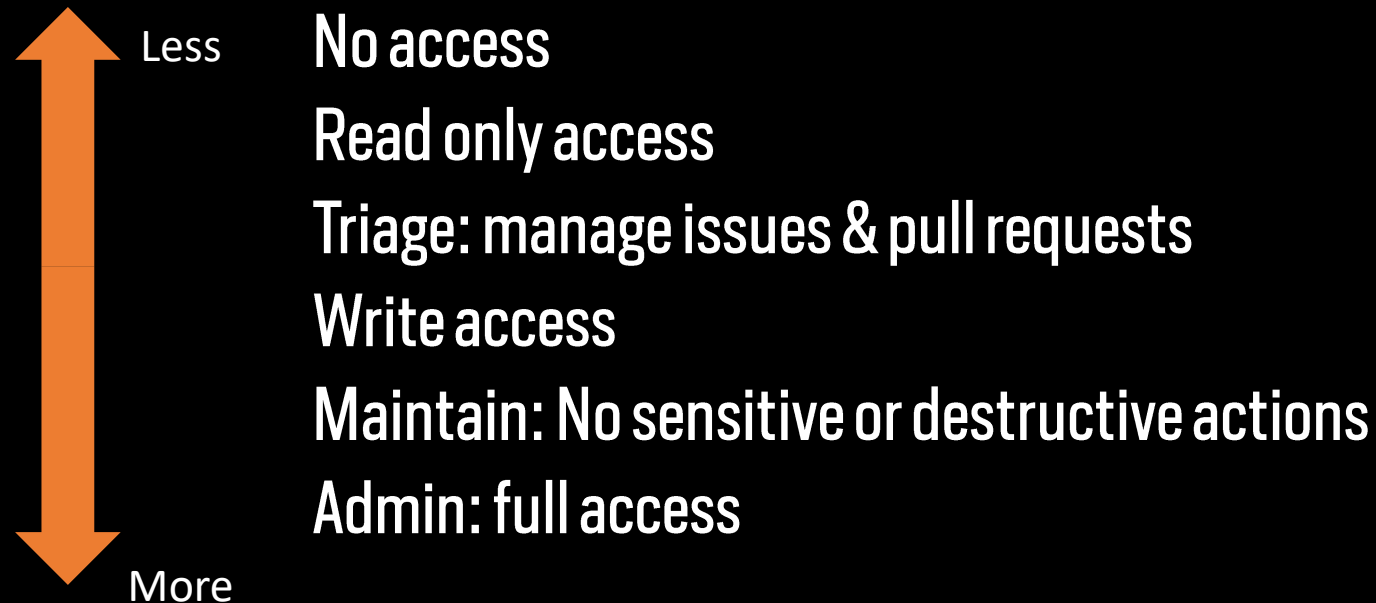
Code – Who has access?

Access levels can be set at:

- Repository
- Organization
- Enterprise

Code – Who has access?

Permission levels



Configuring access

The image displays two screenshots from the GitHub interface, illustrating the steps to configure repository access.

Left Screenshot: Organization Settings

- Organization: GlobalDevOpsBootcamp
- Navigation: Repositories, Packages, **People** (highlighted), Teams, Projects, Insights
- Left sidebar: Organization permissions (Members: 11, Outside collaborators, Pending collaborators, Pending invitations, Failed invitations: 1)
- Search: Find a member...
- Members list (checkboxes for selection):
 - Magnus Kirø
 - Magnus Timner
 - Maxine Chambers
 - Taavi Koosaar
 - mericstam

Right Screenshot: Repository Access

- Repository: GlobalDevOpsBootcamp / PartsUnlimited-Demo2_2020-Team40 (Private)
- Navigation: <> Code, ! Issues, 🔗 Pull requests, ⏮ Actions, 📁 Projects, 📖 Wiki, 🛡 Security, **📊 Insights** (highlighted), ⚙ Settings
- Sub-navigation: **Insights / People** (highlighted)
- Search: Find a user...
- Buttons: Everyone (selected), Outside collaborators, Export CSV
- Message: 11 people have access to this repository
- Access list:
 - Magnus Kirø (magnuskiro) - Read
 - NielsNijveldt - Admin
 - mericstam (mericstam) - Read
 - Marcel de Vries (vriesmarcel) - Admin

From the user

The screenshot shows the GitHub organization page for **GlobalDevOpsBootcamp**. The user **rajbos** (Rob Bos) is highlighted with an orange arrow. The user's role is **Owner**, which is also highlighted with an orange box. The user has access to **81 repositories** and **2 teams**, which is also highlighted with an orange box. The user's membership is **private**, and two-factor security is **enabled**. No SAML identity is linked. The user can be converted to an outside collaborator or removed from the organization.

Owner

As an owner, **rajbos** has **admin access to all repositories** that belong to the GlobalDevOpsBootcamp organization. Manage your owners on the [People page](#).

rajbos has access to 81 repositories

Find a repository they have access to...

Repository	Role	Manage access
GlobalDevOpsBootcamp/PartsUnlimited-Demo2_2020-Team40	Admin on this repository	Manage access
GlobalDevOpsBootcamp/PartsUnlimited-Demo2020-Team03	Admin on this repository	Manage access
GlobalDevOpsBootcamp/PartsUnlimited-Demo2_2020-Team41	Admin on this repository	Manage 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 name

47

publish-profile: \${ secrets.publish_profile } # Define the 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!

Your code – Best practices

Preventing changes by a single person → Pull Request

Protect your main branch

Branch protection rule:

- 1 approver (4 eyes principle)
- Build validation

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 'Settings' tab is highlighted with an orange box. In the left sidebar, the 'Audit log' option is also highlighted with an orange box. The main content area displays the 'Audit log' with a search bar and a list of recent events.

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

23

Your code/repo – trace changes

Account level:

The screenshot shows the GitHub account settings page for user 'rajbos'. The left sidebar contains a list of settings categories: Profile, Account, Appearance (marked 'New'), Account security, Billing & plans, Security log (highlighted with an orange box), Security & analysis, Emails, Notifications, and Scheduled reminders. The main content area is titled 'Security log' and includes a search bar and a 'Filters' dropdown. Below this, the 'Recent events' section lists three actions: 1. 'GitHub System – oauth_authorization.destroy' (Removed authorization for OAuth application was marked as stale (GitHub Co) 9 hours ago), 2. 'rajbos – environment.create_actions_secret' (Created a secret test_env_password for Production 86.93.152.65 | Sint-Michielsgestel, North Brabant, Netherlands | 2 days ago), and 3. 'rajbos – repo.create_actions_secret' (Created a secret for rajbos/dependency-updates 86.93.152.65 | Sint-Michielsgestel, North Brabant, Netherlands | 8 days ago). On the right side, a user menu is open, showing options like 'Set status', 'Your profile', 'Your repositories', 'Your organizations', 'Your enterprises', 'Your projects', 'Your stars', 'Your gists', 'Feature preview', 'Help', 'Settings' (highlighted with an orange box), and 'Sign out'. The top navigation bar includes links for Pull requests, Issues, Codespaces, Marketplace, and Explore, along with a search bar and a notification bell.

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

```
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
```

Self hosted runners

- Starts a HTTPS long poll connection to GitHub asking if there is any work and polls every minute (so outgoing connection)

Can be created at these levels:

- Repository: dedicated to a single repository.
- Organization: multiple repositories in an organization.
- Enterprise: multiple organizations in an enterprise account.

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

Workflows – summary

What happens when someone pushes code into your repository?

Which workflows are triggered? By which branch?

Where are the workflows executed?



GitHub Actions Security

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. The page header includes the GitHub logo, a search bar, and navigation links for Pulls, Issues, Codespaces, Marketplace, and Explore. The breadcrumb trail is 'Marketplace / Actions / EKS on Fargate'. The action is titled 'EKS on Fargate' with the AWS logo and version 'v0.1.1' (Latest version). A green button labeled 'Use latest version' is highlighted with an orange box. Below the title, it says 'WIP: Amazon EKS on AWS Fargate' and 'GitHub Actions'. A description states 'This action allows you to create and manage the lifecycle of an Amazon EKS cluster on AWS Fargate.' It also mentions 'Work in progress, not yet usable.' On the right, it shows 'Verified creator' (aws-actions), 'Stars' (18), and 'Contributors' (aws-actions). An orange arrow points from the 'Use latest version' button to the installation snippet in the modal below.

The screenshot shows the GitHub Actions installation modal for the 'EKS on Fargate' action. It includes the AWS logo, the action name 'EKS on Fargate', and the description 'Creates and EKS on Fargate cluster'. Under the 'INSTALLATION' section, it says 'Copy and paste the following snippet into your .yaml file.' The installation snippet is shown in a code block:

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

 The 'uses' line is highlighted with an orange box. A copy icon is next to the snippet. Below the snippet, it says 'Learn more about this action in aws-actions/amazon-eks-fargate'. An orange arrow points from the 'uses' line to the URL below.

<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

Attack vectors

1. Data Theft
2. Data Integrity Breaches
3. Availability

Protective measures

Manually:

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

Protective measures

```
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

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`. The repository has 4 watchers, 7 stars, and 2 forks. The navigation bar includes links for Code, Issues (2), Pull requests, Actions, Projects, Wiki, Security, and Insights. A blue banner 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 the banner, the repository's main branch is 'main' with 2 branches and 4 tags. A commit by 'tetchel' is shown, titled 'fix os detection bug', with a green checkmark, 7773561 commits, and 40 commits. The commit history shows two entries: '.github/workflows' (13 days ago) and 'tests_/manifests' (2 months ago). The 'About' section on the right describes the GitHub Action as a tool to log in to an OpenShift cluster and set up a Kubernetes context, with a link to the marketplace.

redhat-actions / oc-login

Watch 4 Star 7 Fork 2

Code Issues 2 Pull requests Actions Projects Wiki Security Insights

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 7773561 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/oc-login

openshift kubernetess k8s oc redhat cloud 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.](#)

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 repository page for 'rajbos / dotnetcore-webapp'. The 'Actions' tab is selected, displaying a workflow run titled 'Updating actions with forks (#3) * Update dotnetcore.yml * Update dotnetcore.yml using actions from the rajbos-actions org .NET Core #94'. The workflow status is 'Startup failure'. The 'Summary' section shows the workflow was triggered via push 18 seconds ago. The 'Annotations' section highlights an 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/*.' The error message is enclosed in an orange box. The repository has 110 forks and 0 stars.

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

rajbos pushed c64d658 main

Status: Startup failure

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/*.

.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

Protective measures

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.](#)

Recommendation

- Best practice: Limit to local actions and **fork action repositories**
- Also create a separate org to test actions in
 - Enable DevOps teams to own the actions

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>

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

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.](#)

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

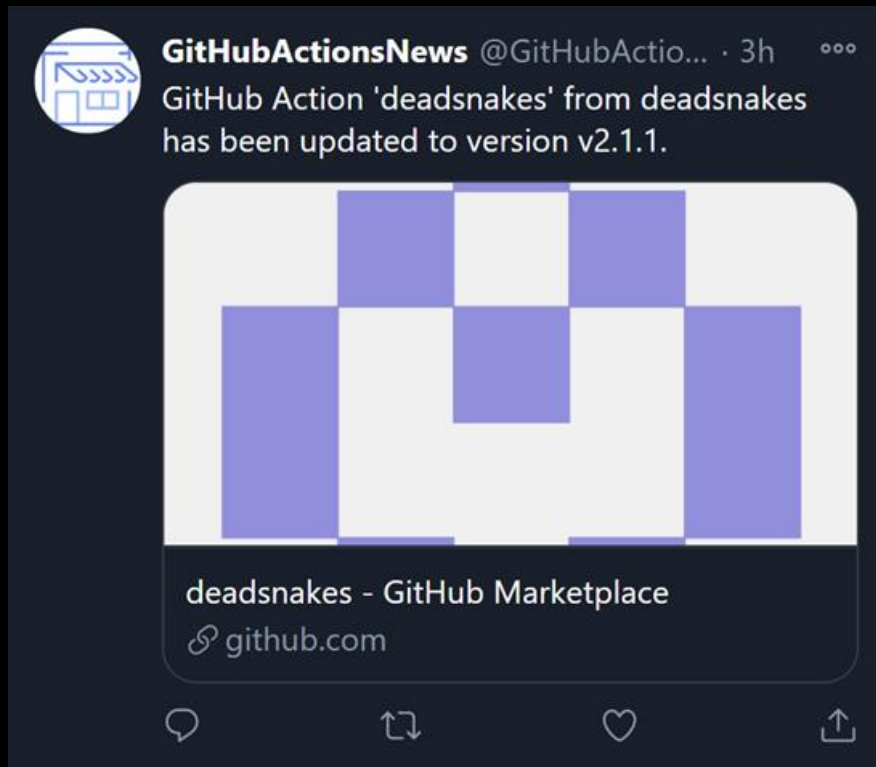
Keeping up to date

Updates

- Actions are updated regularly
 - Wait for a deprecation message?
 - How do you stay up to date?
-
- Auto update with a PR?
 - Read the changes in the source repo

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@44a942b2f7ed0dc101d556f281e906fb791f478

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 targets the `main` branch. The pull request description indicates that `dependabot` wants to merge 1 commit into `main` from `dependabot/github_actions/rajbos-actions/trx-parser-v0.0.5`.

The pull request details show 1 conversation, 1 commit, 3 checks, and 1 file changed. The file `.github/workflows/dotnetcore.yml` is shown with 2 changes. The changes are summarized as `@@ -78,7 +78,7 @@ jobs:`.

The diff shows the following changes in the `jobs:` section:

```
@@ -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 you forked action up to date

The screenshot shows the GitHub interface for a forked repository. At the top, the repository name is 'rajbos-actions / test-repo', with a note 'forked from rajbos/test-repo' highlighted by an orange box. Below the repository name is a navigation bar with links for Code, Pull requests, Actions, Projects, Wiki, and Security. Under the 'Code' tab, there is a 'main' branch selector, a 'Go to file' button, an 'Add file' button, and a green 'Code' button. A large orange box highlights a message stating 'This branch is 2 commits behind rajbos:main.', with links for 'Pull request' and 'Compare'. Below this, a commit history section shows an 'Initial commit' by 'rajbos' from '23 hours ago' with 1 commit. A file named 'README.md' is listed as part of the 'Initial commit' from '23 hours ago'.

Create an update process yourself

Add the parent repo as a remote

Fetch the changes from the parent

Merge in any changes from the branch

Push the changes back to your repo

Gist: <https://xpir.it/UpdateRepo>

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

The screenshot shows a GitHub pull request page for the repository `rajbos / SonarQube-AzureAppService`, which is forked from `vanderby/SonarQube-AzureAppService`. The repository has 1 watch, 0 stars, and 65 forks. The navigation bar includes links 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](#)."

The section "Comparing changes" contains a comparison selector, which is highlighted with an orange box. It shows the following configuration:

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

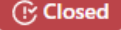
Below the selector, it says "Showing 5 changed files with 283 additions and 44 deletions." The file list shows `.gitignore` with 5 changes. The diff view for `.gitignore` is shown with line numbers 1 through 5. Lines 1, 2, and 3 are unchanged. Lines 4 and 5 are new additions, highlighted in green.

```
...  
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-*/
```


Automation

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

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

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



github-actions  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

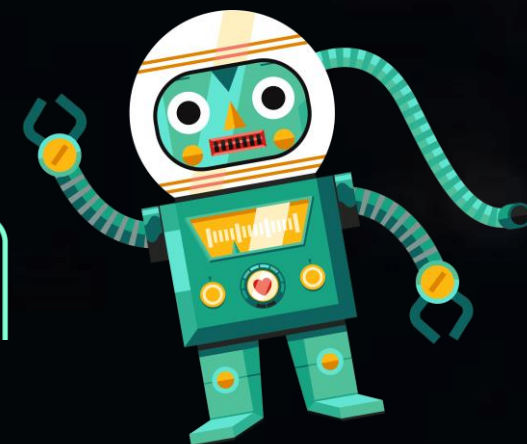
Forking actions
Keeping up to date

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!

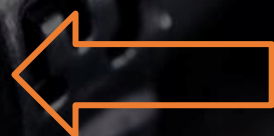
TECHORAMA



Rob Bos

DevOps Consultant - Xpirit

The Netherlands



Q & A at Xpirit Booth!