

GitHub Actions Security

NDC { London }



<https://myoctocat.com>

GitHub Actions Security

NDC { London }

Rob Bos

DevOps Consultant – Xpirit

The Netherlands

@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 Codespaces Marketplace Explore

Notifications + Account

Marketplace / Search results

Types

Search for apps and actions

Apps

Actions

X

Categories

API management



Codefresh pipeline runner

By codefresh-io ✅

Github action that runs codefresh pipeline
43 stars

Chat

Code quality

Code review

Continuous integration

Dependency management

Deployment

IDEs

Learning

Localization

Actions

An entirely new way to automate your development workflow.

6743 results

Filtered by Actions

X

Codefresh pipeline runner

By codefresh-io ✅

Github action that runs codefresh pipeline
43 stars



Anchore Container Scan

By anchore ✅

Scan docker containers with Gype for
vulnerabilities
59 stars



DefenseCode ThunderScan Action

By defensecode ✅

Source code scanning for vulnerabilities
using DefenseCode ThunderScan SAST
solution
14 stars



Jira Find issue key

By atlassian ✅

Find an issue inside event
12 stars



Jira Add Comment

By atlassian ✅

Add a comment to an issue
7 stars



calibreapp/github-actions

By calibreapp ✅

Wraps the Calibre CLI to enable common
commands
5 stars

Workflow example

The screenshot shows a GitHub repository named "dotnetcore-webapp" with a workflow configuration file named "dotnetcore.yml". The file is displayed in a code editor with line numbers on the left. Several sections of the code are highlighted with orange boxes:

- on: [push]**: A section under the "jobs" key.
- steps:**: A key under the "build-and-deploy" job.
- uses: actions/checkout@v1**: An item under the "steps" key.
- name: Setup .NET Core**: A key under the "steps" key.
- uses: actions/setup-dotnet@v1**: An item under the "steps" key.
- with:**: A key under the "steps" key.
- dotnet-version: 3.0.100**: A value under the "with" key.
- # dotnet build**: A comment at the start of a new section.
- name: Build with dotnet**: A key under the "steps" key.
- run: |**: A key under the "steps" key.
- dotnet build --configuration Release ./dotnet-core-webapp/dotnetcore-webapp.csproj**: A value under the "run" key.

```
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   - name: Setup .NET Core
14     uses: actions/setup-dotnet@v1
15     with:
16       dotnet-version: 3.0.100
17
18 # dotnet build
19 - name: Build with dotnet
20   run: |
21     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?

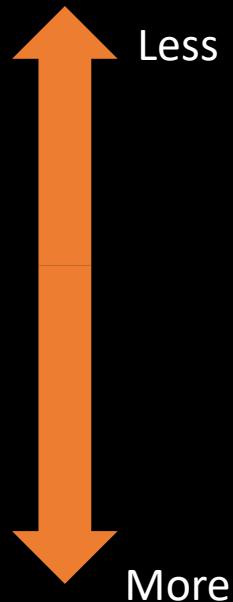
Access levels can be set at:

- 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

Configuring access

The screenshot shows two views of GitHub's access configuration interface.

Left View: Organization Permissions

- Header: GlobalDevOpsBootcamp
- Navigation: Repositories, Packages, **People**, Teams, Projects, Insights
- Section: Organization permissions
- Members: 11 (Magnus Kirø, Magnus Timmer, Maxine Chambers, Taavi Koosaar, mericstam)
- Outside collaborators
- Pending collaborators
- Pending invitations
- Failed invitations: 1 (mericstam)

Right View: Repository Access

- Header: GlobalDevOpsBootcamp / PartsUnlimited-Demo2_2020-Team40 (Private)
- Navigation: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, **Insights**, Settings
- Section: Insights / People
- Search: Find a user... (Everyone, Outside collaborators)
- Text: 11 people have access to this repository
- Table of users with access levels:

User	Access Level
Magnus Kirø (magnuskirø)	Read
NielsNijveldt	Admin
mericstam (mericstam)	Read
Marcel de Vries (vriesmarcel)	Admin

From the user

The screenshot shows the GitHub organization settings page for 'GlobalDevOpsBootcamp'. The top navigation bar includes links for Repositories, Packages, People (which is the active tab), Teams, Projects, Insights, and Settings. On the left, a sidebar for the owner 'rajbos' (Rob Bos) displays statistics: 81 repositories and 2 teams. It also shows that membership is private, two-factor security is enabled, and no SAML identity is linked. Buttons for 'Convert to outside collaborator' and 'Remove from organization' are at the bottom. The main content area is titled 'Owner' and states that 'rajbos has admin access to all repositories that belong to the GlobalDevOpsBootcamp organization'. A search bar allows finding specific repositories. Three repository entries are listed, each showing 'Admin on this repository' and a 'Manage access' button. The repositories are: 'PartsUnlimited-Demo2_2020-Team40', 'PartsUnlimited-Demo2020-Team03', and 'PartsUnlimited-Demo2_2020-Team41'. An orange arrow points to the 'Owner' section, and another orange box highlights the repository statistics in the sidebar.

GlobalDevOpsBootcamp

Repositories Packages People Teams Projects Insights Settings

rajbos Rob Bos

Owner

81 repositories 2 teams

Membership private ▾ Two-factor security enabled No SAML identity linked

Convert to outside collaborator Remove from 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...

Manage access ?

Admin on this repository

Manage access ?

Admin on this repository

Manage access ?

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

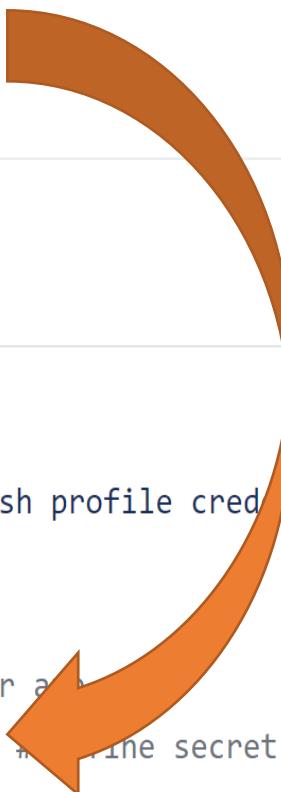
 SONAR_TOKEN

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'
```



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

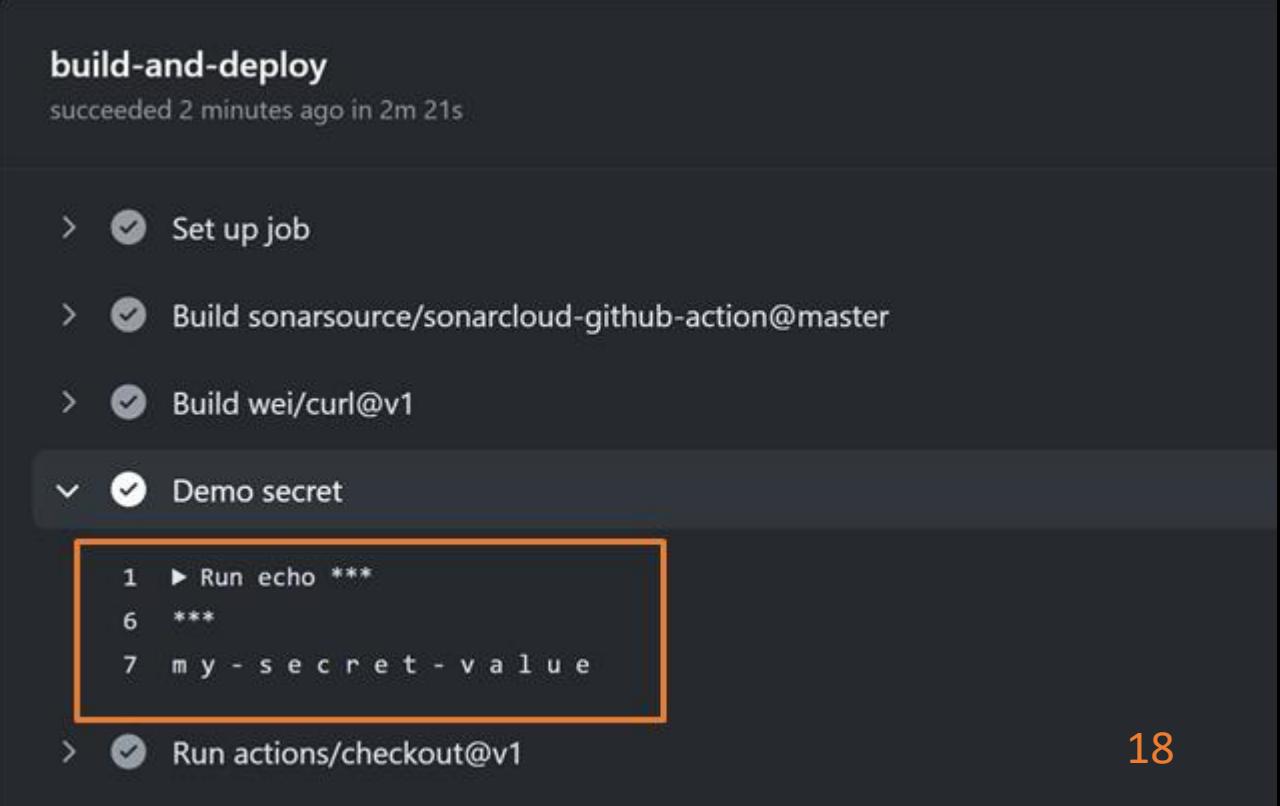
Will be redacted in logs

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

Who has access to your secrets?

- The Action 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 my-secret-value  
  
> ✓ 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:

- Linters
- 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 left sidebar lists various organization settings like Profile, Billing & plans, Member privileges, etc., with 'Audit log' highlighted by an orange box. The right side 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. The bottom right corner of the slide has the number '23'.

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 interface. On the left, a sidebar lists account management options: Profile, Account, Appearance (New), Account security, Billing & plans, Security log (which is highlighted with an orange box), Security & analysis, Emails, Notifications, and Scheduled reminders. On the right, the 'Security log' section displays a list of recent events. The first event is from the GitHub System, showing the removal of an OAuth authorization. The second event is from 'rajbos', creating a secret named 'test_env_password' for the 'Production' environment. The third event is also from 'rajbos', creating a secret for the repository 'rajbos/dependency-updates'. The top navigation bar includes links for Pull requests, Issues, Codespaces, Marketplace, and Explore. The top right corner shows the user is signed in as 'rajbos' and provides a dropdown menu with options like Set status, Your profile, Your repositories, etc., with 'Settings' also highlighted with an orange box.

Signed in as **rajbos**

Profile

Account

Appearance New

Account security

Billing & plans

Security log

Filters ▼ Search your security log

Recent events

GitHub System – oauth_authorization.destroy
Removed authorization for OAuth application was marked as stale (GitHub C
9 hours ago

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

rajbos – repo.create_actions_secret
Created a secret for [rajbos/dependency-updates](#)
86.93.152.65 | Sint-Michielsgestel, North Brabant, Netherlands | 8 days ago

Set status

Your profile

Your repositories

Your organizations

Your enterprises

Your projects

Your stars

Your gists

Feature preview

Help

Settings

Sign out

GitHub Actions Security

- Repository security
 - Runners and security
 - Actions and security
-
- Forking actions
 - Keeping up to date



Workflow Runners

Actions execute on runners

GitHub hosted

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

Self hosted

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

```
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
```

```
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:
```

Self-hosted runners

<https://github.com/actions/runner>

Download

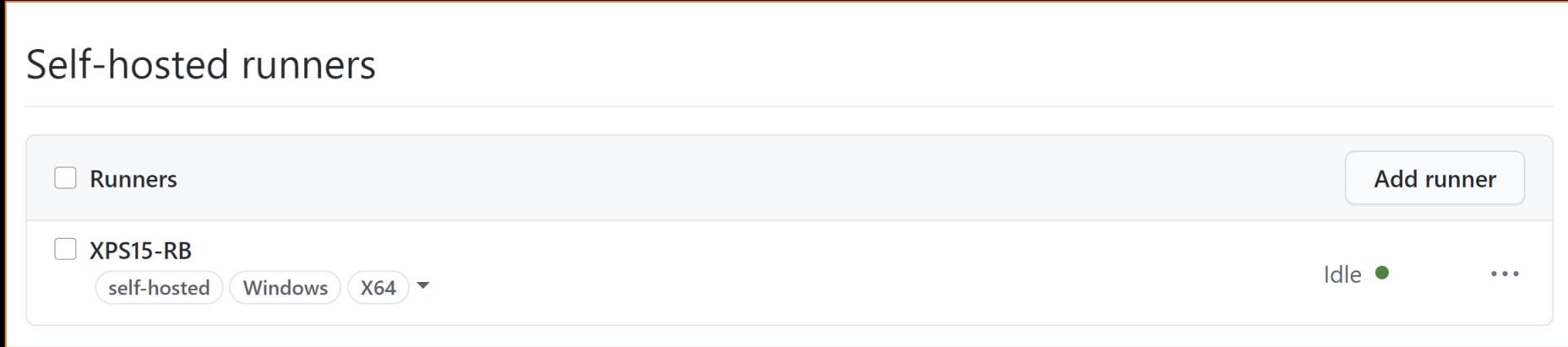
We recommend configuring the runner under "\actions-runner". This will help avoid issues related to service identity folder permissions and long path restrictions on Windows.

```
# Create a folder under the drive root  
$ mkdir actions-runner; cd actions-runner  
  
# Download the latest runner package  
  
$ Invoke-WebRequest -Uri https://github.com/actions/runner/releases/download/v2.275.1/actions-runner-win-x64-2.275.1.zip -OutFile  
actions-runner-win-x64-2.275.1.zip  
  
# Extract the installer  
  
$ Add-Type -AssemblyName System.IO.Compression.FileSystem ; [System.IO.Compression.ZipFile]::ExtractToDirectory("$PWD/actions-  
runner-win-x64-2.275.1.zip", "$PWD")
```

Configure

```
# Create the runner and start the configuration experience  
$ ./config.cmd --url https://github.com/rajbos/dotnetcore-webapp --token ABONY4I[REDACTED]OBQG  
  
# Run it!  
$ ./run.cmd
```

Self hosted runners



Automatically updates itself when a new workflow is started or within a week after a new release

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

- To what parts of your environment could a runner potentially have access?
- Think of network and shared state between runs
- User: limit its access! (don't run as administrator / root when not needed)

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:

<http://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

Workflows

What happens when someone pushes code into your repository?

Which workflows are triggered? By which branch?



GitHub Actions Security

Repository security

Runners and security

Actions and security

Forking actions

Keeping up to date

Actions and security



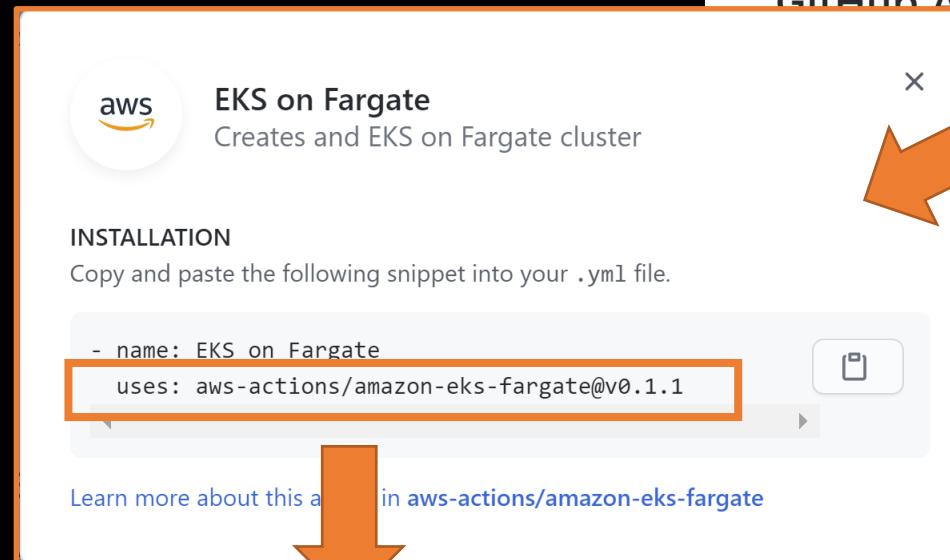
Are you running just any action from the internet?



SCARY, especially in an Enterprise or on local runners

Actions

- Marketplace or by direct url



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

A screenshot of the GitHub Actions page for the 'EKS on Fargate' action. The page shows the action's details, including its logo (aws), name (EKS on Fargate), version (v0.1.1), and status (Latest version). A large orange arrow points from the 'Use latest version' button on the right to the 'Latest version' badge on the left. The page also includes sections for 'WIP: Amazon EKS on AWS Fargate GitHub Actions', 'DESCRIPTION', 'INSTALLATION', and 'STARS'.

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 top navigation bar includes 'Watch' (4), 'Star' (7), 'Fork' (2), and tabs for 'Code', 'Issues (2)', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', and 'Insights'. A prominent callout box highlights the 'Actions' tab, which contains the text: 'Use this GitHub Action with your project. Add this Action to an existing workflow or create a new one.' with a 'View on Marketplace' button. Below this, the repository details show 'main' branch, 2 branches, 4 tags, and a commit history with the latest being 'tetchel fix os detection bug' by 'tetchel' 10 days ago. The bottom right corner features a sidebar with links to the marketplace listing and various tags: openshift, kubernetes, k8s, oc, redhat, cloud, and action.

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

openshift kubernetes 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 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 job is listed:

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

The job summary table includes:

Triggered via push 18 seconds ago	Status	Total duration	Artifacts
<code>rajbos pushed -o c64d658 main</code>	Startup failure	-	-

The **Annotations** section shows one 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, before forking them
 - To enable DevOps teams to have the autonomy to test and verify themselves

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)

Enable DevOps teams to test 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](#)

Be aware: setting this on the organization, will prevent any repository in that org to change that setting to all actions.

Create a test org so teams can test other actions to prevent them from being completely blocked.



GitHub Actions Security

Repository security

Runners and security

Actions and security

Forking actions

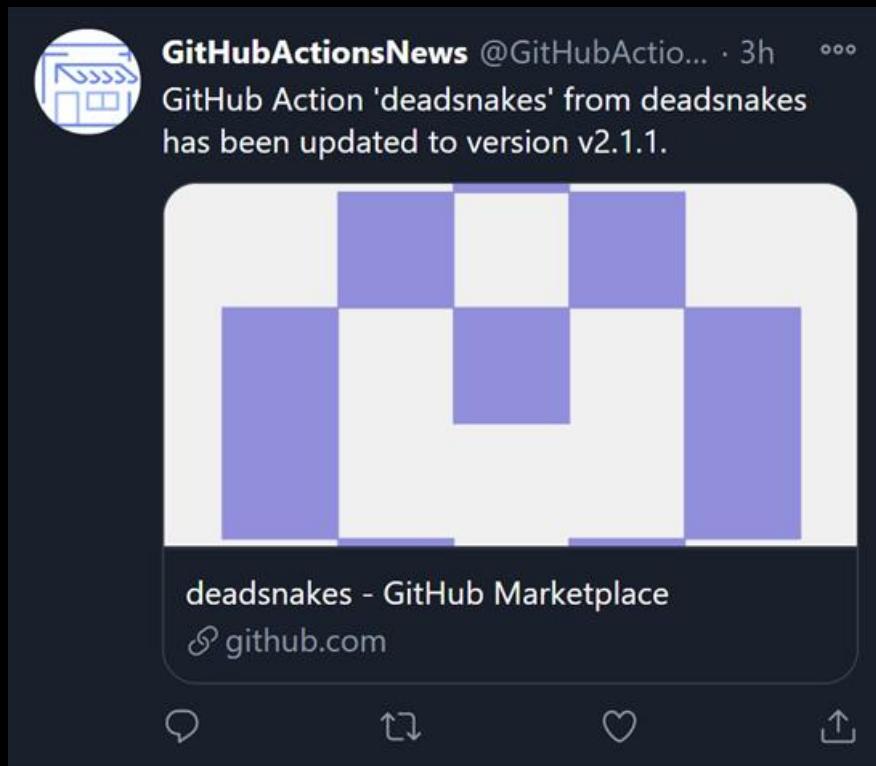
Keeping up to date

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!



Use Dependabot

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 summary indicates that dependabot wants to merge 1 commit from `dependabot/github_actions/rajbos-actions/trx-parser-v0.0.5`. The changes are shown in the `.github/workflows/dotnetcore.yml` file, where the `uses` key for the `trx-parser` action is updated from `v0.0.3` to `v0.0.5`.

```
diff --git a/.github/workflows/dotnetcore.yml b/.github/workflows/dotnetcore.yml
index 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          TRX_PATH: ${{ github.workspace }}\\dotnet-core-webapp.webtests\\TestResults #This should be the path to your TRX files
```

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 tab is selected, showing the `main` branch. A message indicates that the branch is 2 commits behind the `rajbos:main` branch. The commit history shows two recent commits: one from `rajbos` and another for `README.md`.

Key elements visible on the page:

- Repository name: `rajbos-actions / test-repo`
- Forked from: `rajbos/test-repo`
- Branch: `main`
- Status message: "This branch is 2 commits behind rajbos:main."
- Actions: Pull request, Compare
- Commits:
 - `rajbos Initial commit` (23 hours ago)
 - `README.md Initial commit` (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: <http://xpir.it/UpdateRepo>

Automate the update

Use a workflow!

<https://github.com/aormsby/Fork-Sync-With-Upstream-action>

Downsides:

- Needs to be added to each fork and on a new branch
- Workflows on non-default branch issue
- You're still pulling in all changes: how do you handle reviews?

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 main heading reads: "Parent repository for [rajbos/SonarQube-AzureAppService] has updates available #25". Below this, it says "github-actions (bot) opened this issue 22 hours ago · 0 comments". A comment from the same bot follows, stating: "The parent repository for `rajbos/SonarQube-AzureAppService` has updates available." It includes a warning: "Important! Click on this [compare link](#) to check the incoming changes before updating the fork." and instructions: "To update the fork Add the label `update-fork` to this issue to update the fork". On the right side, there are sections for "Assignees" (None yet), "Labels" (None yet), "Projects" (None yet), and "Milestone" (None yet). The top navigation bar includes links for Search or jump to..., Pulls, Issues, Codespaces, Marketplace, Explore, and user profile icons.

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

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 None—assign yourself

Labels None yet

Projects None yet

Milestone None

Review before merging

Compare url:

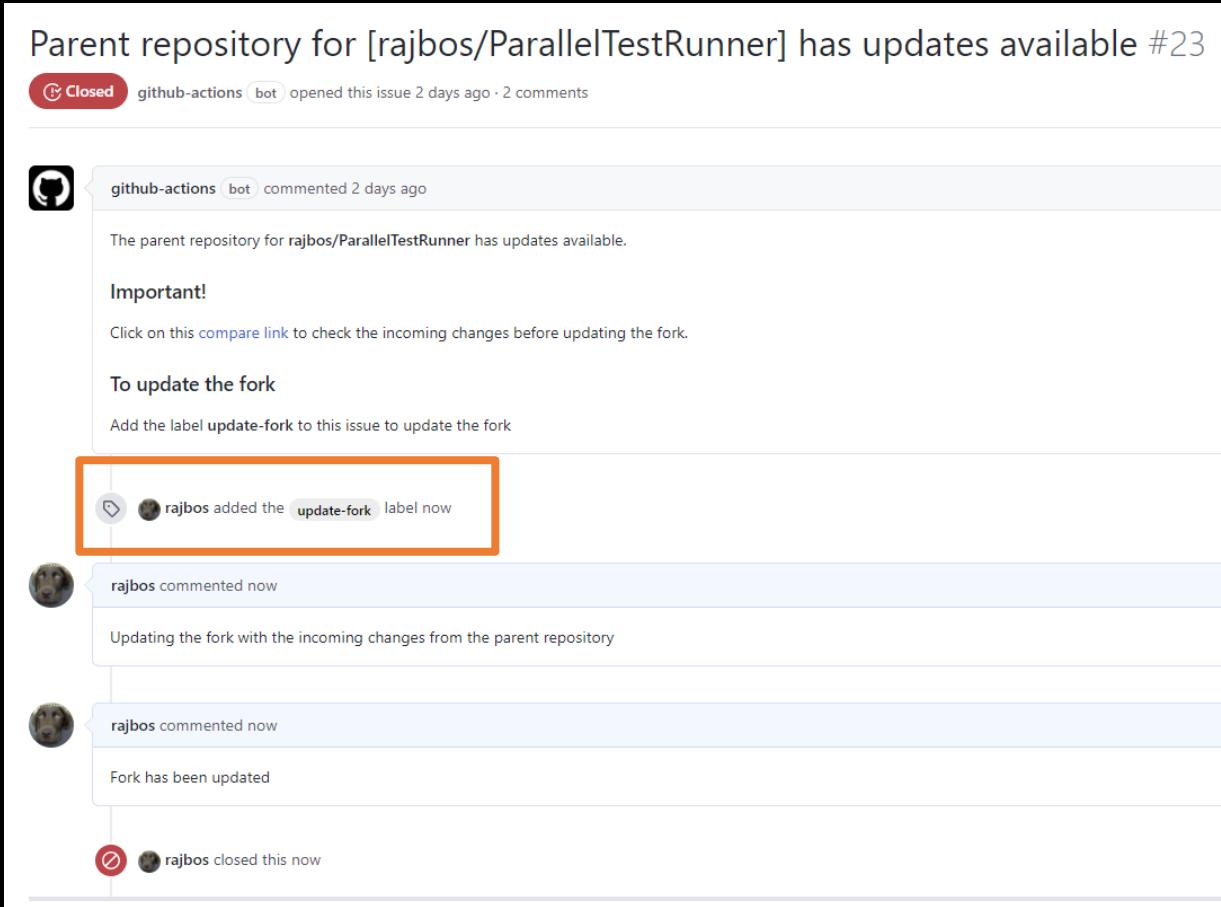
<https://github.com/rajbos-actions/test-repo/compare/main...rajbos:main>

The screenshot shows a GitHub repository page for `rajbos/SonarQube-AzureAppService`, which is a fork of `vanderby/SonarQube-AzureAppService`. The page displays a comparison between the `main` branch and the `master` branch. The comparison interface includes dropdowns for selecting the base repository (`rajbos/SonarQube-AzureAppS...`) and base branch (`master`), and the head repository (`vanderby/SonarQube-AzureAp...`) and compare branch (`master`). Below the dropdowns, a message indicates that 5 changed files have 283 additions and 44 deletions. A preview of the changes in the `.gitignore` file is shown, highlighting a new line that excludes the `sonarqube-*/` directory.

```
Showing 5 changed files with 283 additions and 44 deletions.  
Unified Split  
8 .gitignore  
... ... @@ -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-*/
```

Automation

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



GitHub Actions Security

Repository security

Runners and security

Actions and security

Forking actions

Keeping up to date

Best practices summarized

- Set access levels per team, not per user
- Treat workflow secrets very carefully: best to think of them as public
- Review actions' source code
- Pin actions to commit SHA
- 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!

Rob Bos

DevOps Consultant - Xpirit

The Netherlands

NDC { London }