Characterizing the Security of GitHub CI Workflows

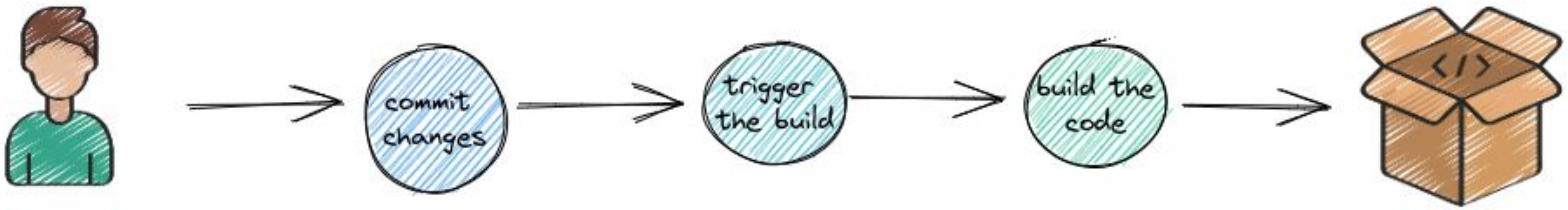
Igibek Koishybayev[†], Aleksandr Nahapetyan[†], Raima Zachariah[§], Siddharth Muralee[‡], Bradley Reaves[†], Alexandros Kapravelos[†], Aravind Machiry[‡]

[†]North Carolina State University, [‡]Purdue University, [§]Independent Researcher

NC STATE UNIVERSITY

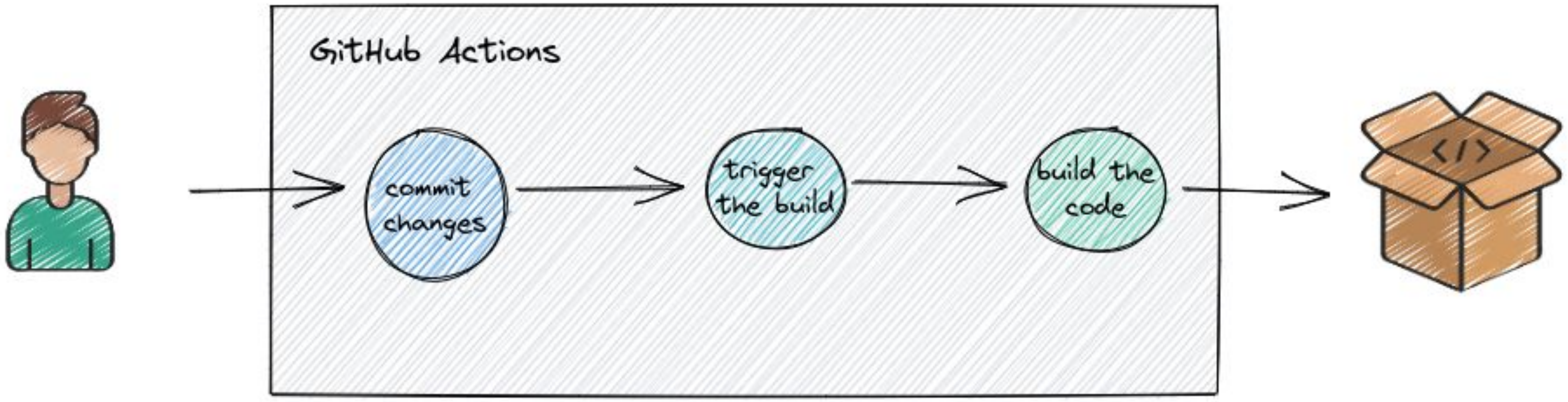
PURDUE UNIVERSITY

Intro to CI/CD



CI/CD is part of software development supply-chain and requires at least the same level of secure management as a final product

What is GitHub Actions?



Features:

- supports third-party plugins
- allows to self-host the runner
- manages the secrets

GitHub Actions gained tremendous popularity in usage among OSS

Workflow Details

```
name: "Build and Test workflow"
on: [push, pull_request]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: "Setup PHP"
        uses: shivammathur/setup-php@master
        with:
          php-version: "8.1"
      - run: composer install
      - name: "Codecov"
        uses: codecov/codecov-action@29386c70e*
        with:
          token: ${{ secrets.CODECOV_TOKEN }}
```

Triggers

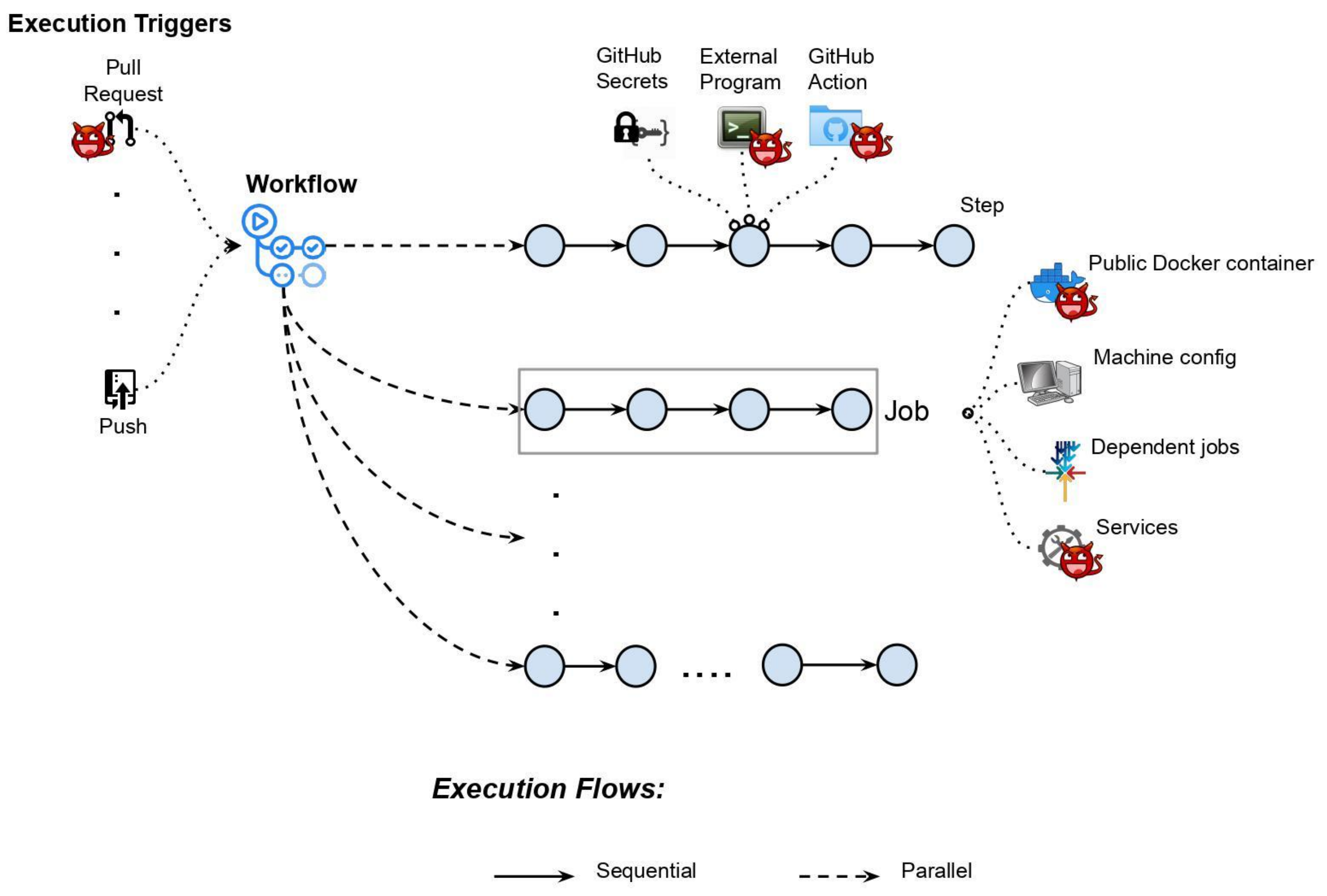
Job’s name

Job contains multiple steps

References third-party plugins

Passing the secret to Action

.github/workflows/build.yml



Research Questions

1. What are the security properties that need to be hold to have a secure CI/CD?

2. How does GitHub Actions compare to other CI/CD platforms according to SPs?

3. How does usage behavior of workflows affect GitHub Actions SPs?

Security Properties

- **Admittance Control**
  - only the people with the right permissions must be able to add, delete, or modify workflows to the repository
- **Execution Control**
  - only authorized users must be able to configure the events that trigger the execution of workflow
- **Code Control**
  - which code can run as part of the workflow
- **Access to Secrets**
  - ensure that secrets can be accessed by only those steps to which secret is explicitly passed

Compare GitHub Actions Default Permissions with Others

CI/CD Platforms	**Permissions**	
	Code read	**Code write**
TravisCI	🟢	◐
CircleCI	🟢	◐
Jenkins	🟢	🔴
Gitlab CI (external)	🟢	◐
Gitlab CI (internal)	🟢	○
Github Actions	🟢	🔴

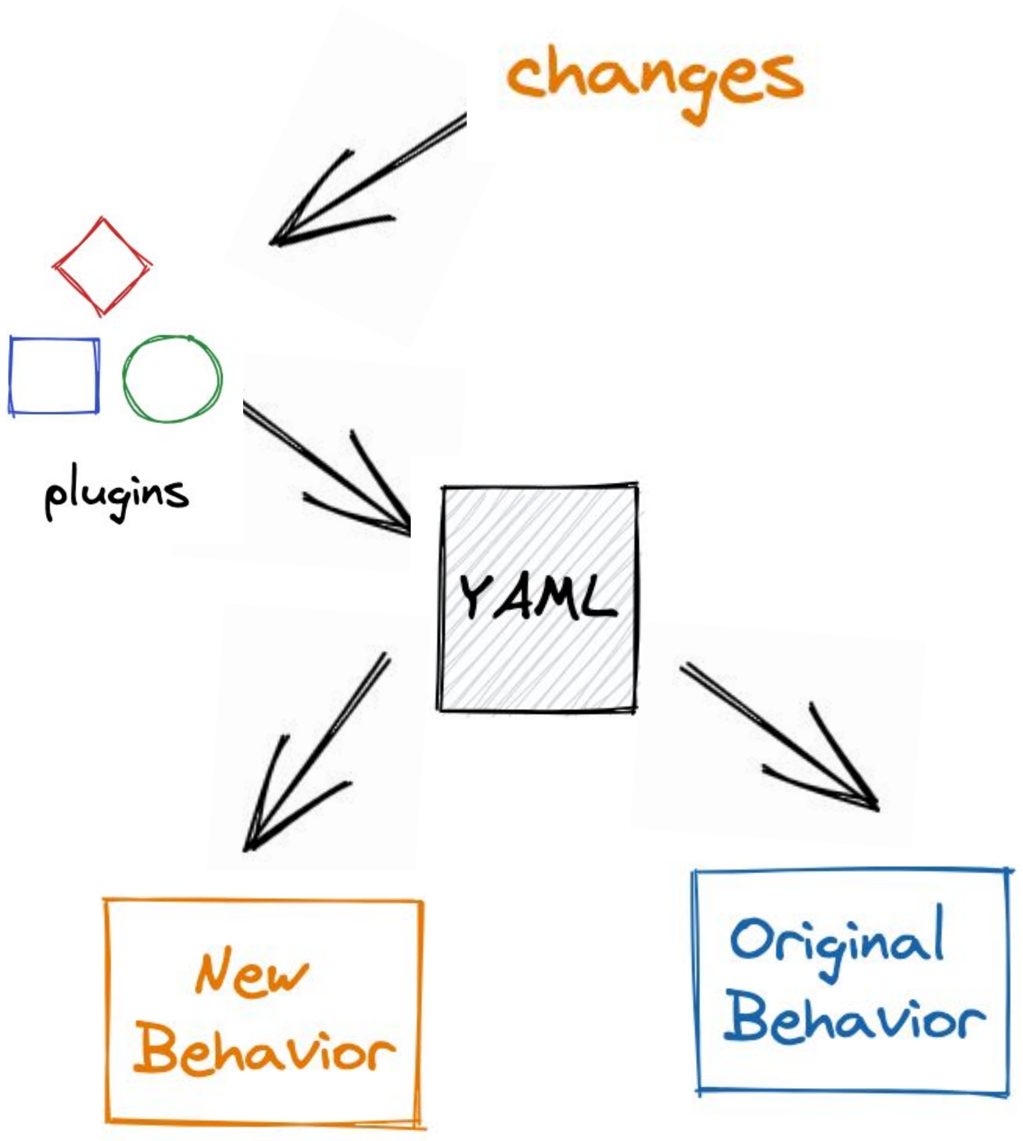
Default code read & write permissions of different CI/CD platform. **Red** color means the “bad” behavior, while **green** color means the “good” behavior

Additionally, all steps in GitHub workflow runs with **administrator** privileges

Compare GitHub Actions Plugin System with Others

CI/CD Platforms	Plugins			
	First-party	Third-party	Mutable	Review
TravisCI	🟢	◐	○	⭕
CircleCI	🟢	🔴	○	⭕
Jenkins	○	🔴	○	⭕
Gitlab CI (external)	🟢	○	○	⭕
Gitlab CI (internal)	🟢	○	○	⭕
Github Actions	🟢	🔴	🔴	⭕

Plugin support by different CI/CD platforms. **Red** color means the “bad” behavior, while **green** color means the “good” behavior



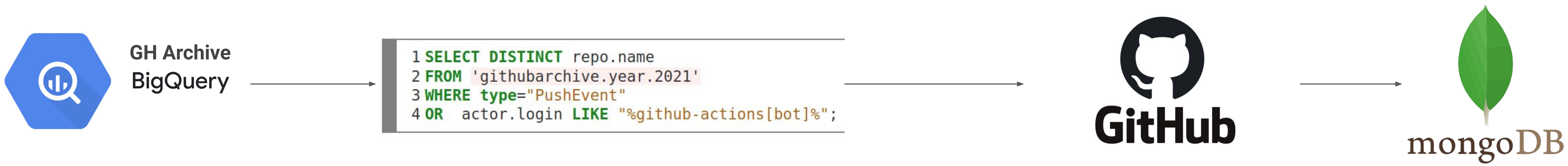
Security Properties Comparison Between GitHub CI and Others

		TravisCI	CircleCI	Jenkins	Gitlab CI (external)	Gitlab CI (internal)	GitHub Actions
Admittance Control	(C1) Contributors can add a new workflow	🟢	🟢	🟢	🟢	🟢	🟢
	(C2) CI/CD run can add a new workflow	○	○	🔴	○	○	🔴
	(C3) Executes workflow from PR w/o merge	○	◐	🔴	○	◐	◐
Execution Control	(C4) Contributors can modify the triggers	○	○	○	🟢	🟢	🟢
	(C5) CI/CD run can modify the triggers	○	○	🔴	○	○	🔴
Code Control	(C6) CI/CD run can modify the code	○	○	🔴	○	○	🔴
	(C7) CI/CD run can change the behavior w/o modifying the config	○	○	○	○	○	🔴
Access to Secret	(C8) Masked	🟢	🟢	◐	🟢	🟢	🟢
	(C9) Available to all steps	🔴	🔴	◐	○	○	🔴
	(C10) Available to pull requests	○	◐	◐	○	◐	◐

Comparison of five different CI/CD platforms in four different security properties. **Red** color means the "bad" behavior, while **green** color means the "good" behavior

All steps can read **/home/runner/_work** and access the secrets without direct access to secret

Large-Scale Measurement Experiment



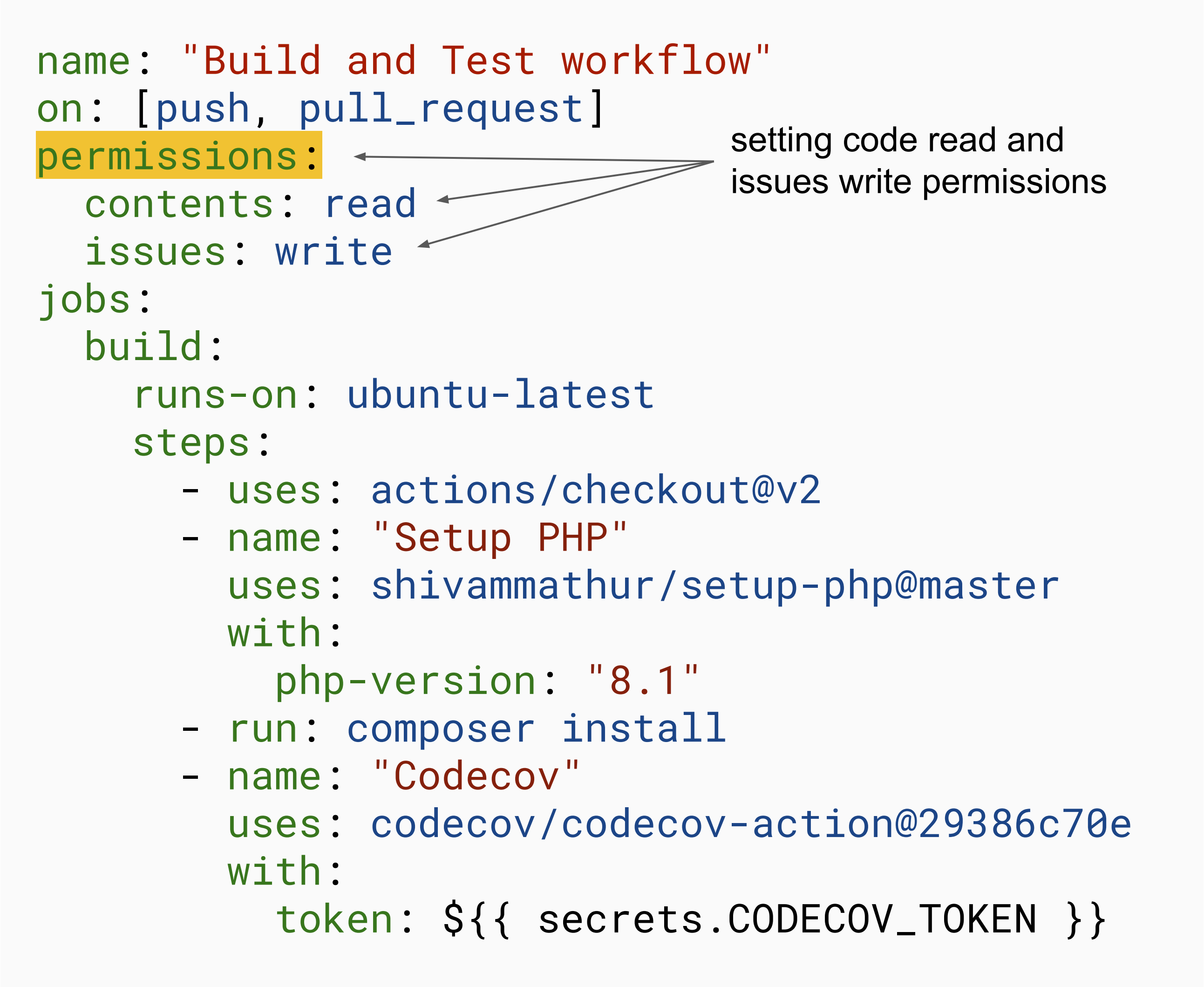
```sql
SELECT DISTINCT repo.name
FROM 'githubarchive.year.2021'
WHERE type="PushEvent"
OR  actor.login LIKE "%github-actions[bot]%";
```

In total 213,854 public repos
with 447,238 workflows

Repository: https://github.com/wspr-ncsu/github-actions-security-analysis

Q1: Do developers update default permissions?

- Only **900**/447K or **0.2%** workflows customize permissions
  - Among them 62% to read-only



```
name: "Build and Test workflow"
on: [push, pull_request]
permissions:
  contents: read
  issues: write
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: "Setup PHP"
        uses: shivammathur/setup-php@master
        with:
          php-version: "8.1"
      - run: composer install
      - name: "Codecov"
        uses: codecov/codecov-action@29386c70e
        with:
          token: ${{ secrets.CODECOV_TOKEN }}
```

setting code read and issues write permissions

Q2: How GitHub workflows are triggered and are the triggers used in dangerous ways?

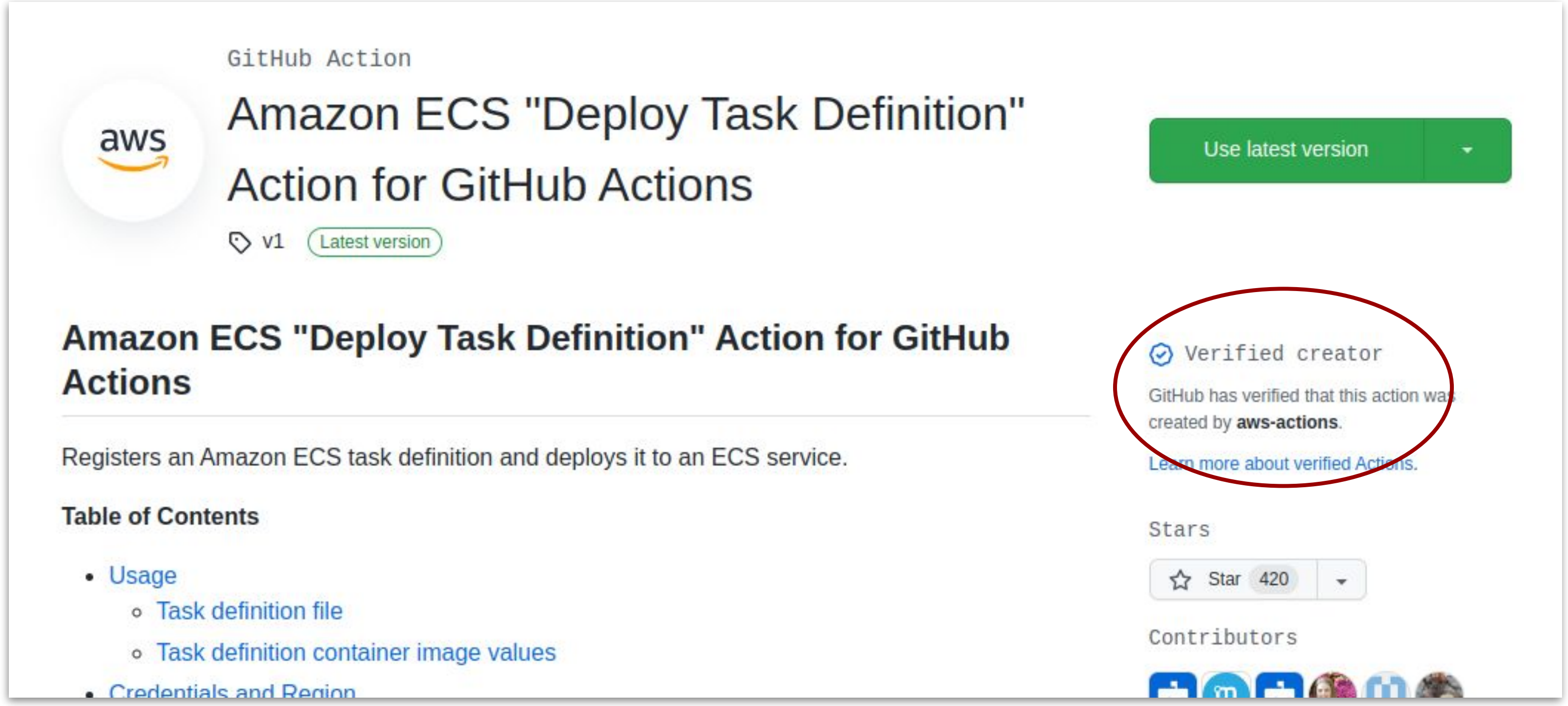
- It is *possible** to introduce new workflows through PRs
- At least **292** repos with `pull_request` workflow(s) are self-hosted machine
  - TLDR; execute arbitrary code on your machine with pull request

Trigger events	Repositories (%)	Workflows (%)
`push`	179,503 (83.9%)	279,337 (62.5%)
`pull_request`	94,962 (44.4%)	146,803 (32.8%)
`cron`	51,544 (24.1%)	70,719 (15.8%)
`manual`	45,134 (21.1%)	83,616 (18.7%)
`pull_request_target`	7,485 (3.5%)	8,874 (1.9%)

Repositories with at least one workflow triggered on **`push`**, **`pull_request`**, **`cron`**, **`manual`**, and **`pull_request_target`** events

Q3: Do users depend on third-party plugins?

- **99.7%** of repositories uses third-party actions
- Overall 11,438 unique actions w/o version are used
  - Overall 19,033 unique actions w/ version are used
- Only **335** (**2.9%**) out of 11,438 of actions are from **verified creators**



GitHub Action

Amazon ECS "Deploy Task Definition" Action for GitHub Actions

v1 Latest version

Use latest version

Amazon ECS "Deploy Task Definition" Action for GitHub Actions

Registers an Amazon ECS task definition and deploys it to an ECS service.

Table of Contents

- Usage
  - Task definition file
  - Task definition container image values
- Credentials and Region

Verified creator

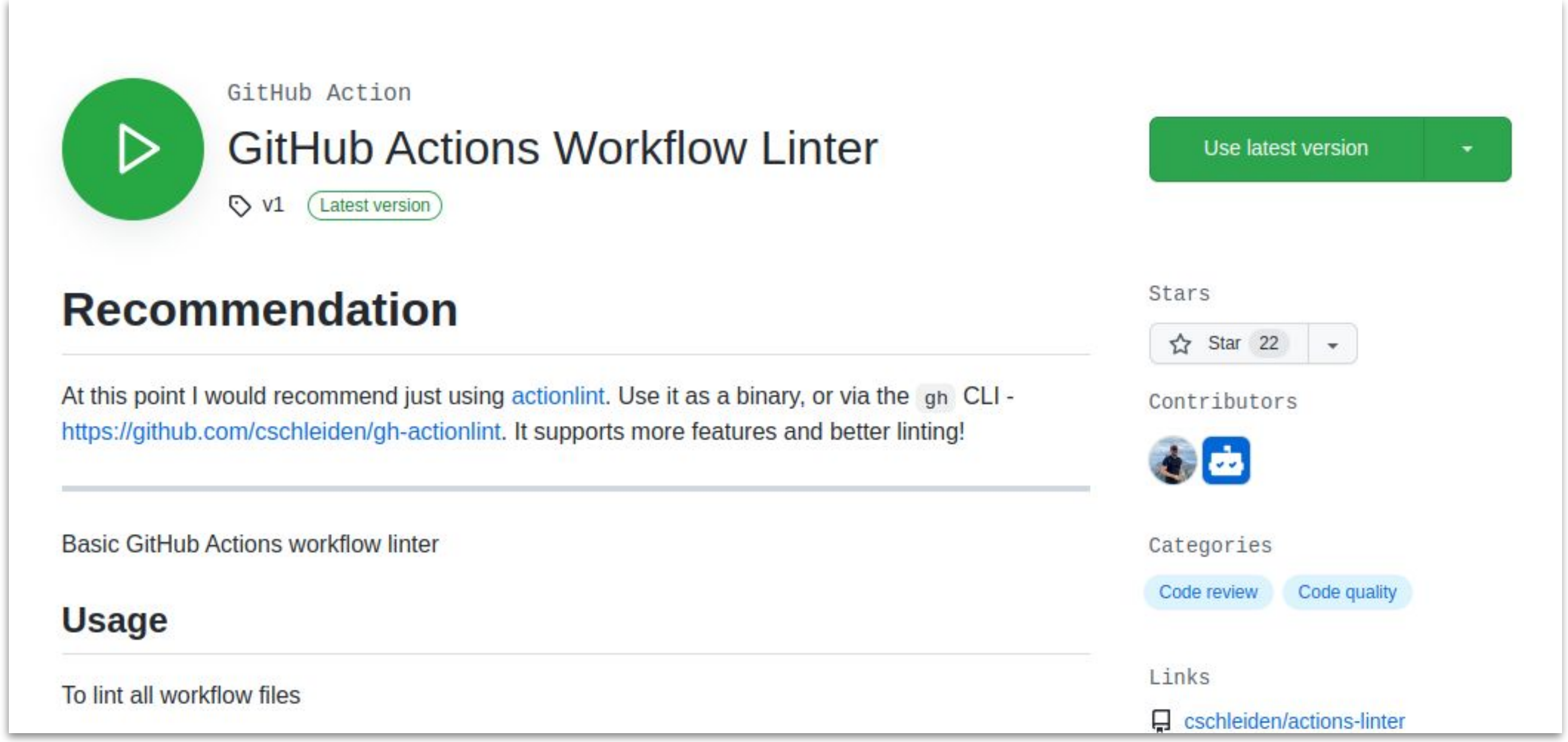
GitHub has verified that this action was created by **aws-actions**.

Learn more about verified Actions.

Stars

Star 420

Contributors



GitHub Action

GitHub Actions Workflow Linter

v1 Latest version

Use latest version

Recommendation

At this point I would recommend just using actionlint. Use it as a binary, or via the `gh` CLI - https://github.com/cschleiden/gh-actionlint. It supports more features and better linting!

Basic GitHub Actions workflow linter

Usage

To lint all workflow files

Stars

Star 22

Contributors

Categories

Code review Code quality

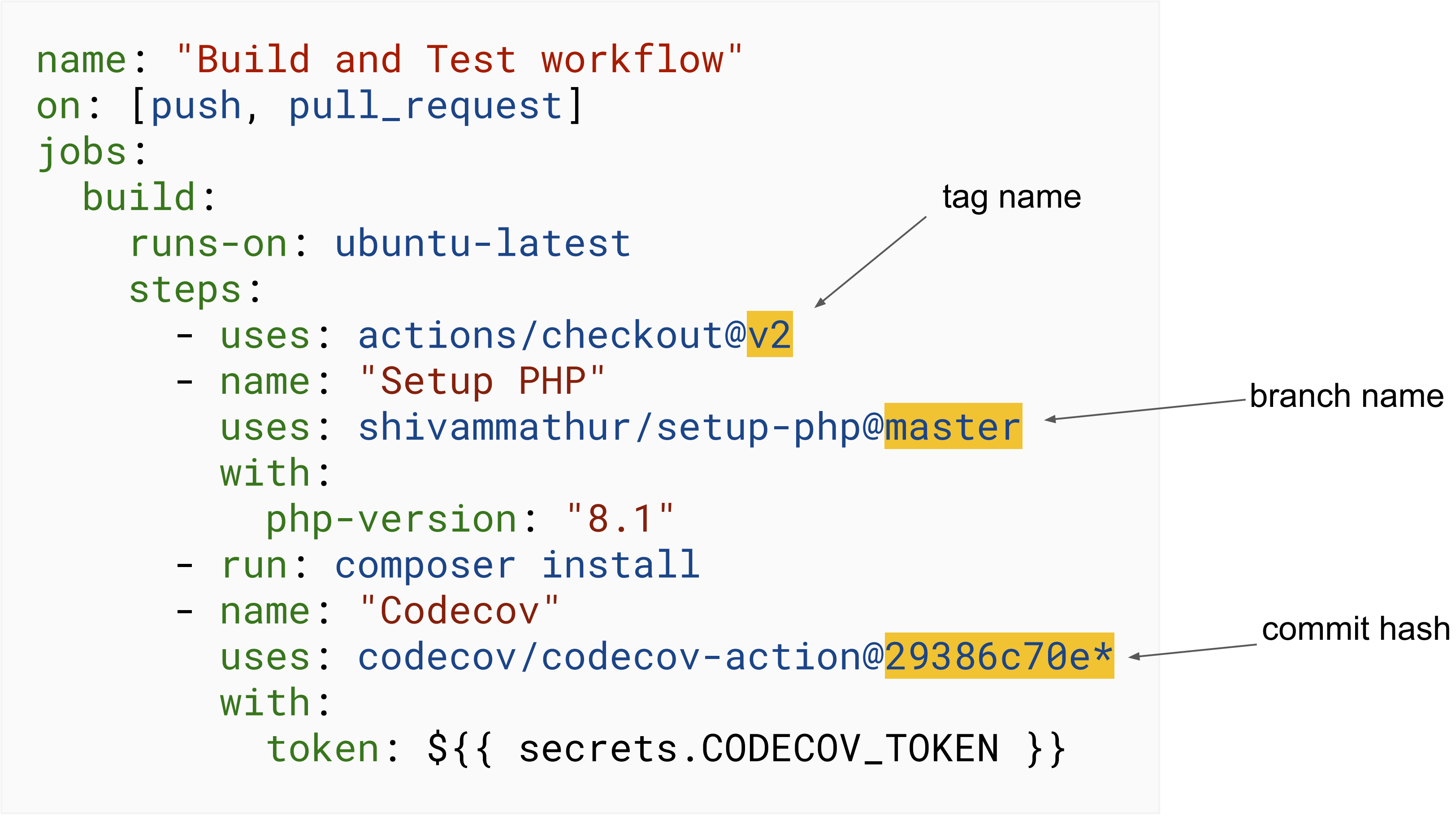
Links

cschleiden/actions-linter

Q4: How users reference third-party plugins?

Reference types	References (non-verified)
Tag name	474,166 (410,054)
Branch name	120,633 (109,400)
Commit hash	6,539 (5,687)

Distribution of third-party actions reference types. Only **0.1%** of references are commit hash (aka immutable)



```
name: "Build and Test workflow"
on: [push, pull_request]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: "Setup PHP"
        uses: shivammathur/setup-php@master
        with:
          php-version: "8.1"
      - run: composer install
      - name: "Codecov"
        uses: codecov/codecov-action@29386c70e*
        with:
          token: ${{ secrets.CODECOV_TOKEN }}
```

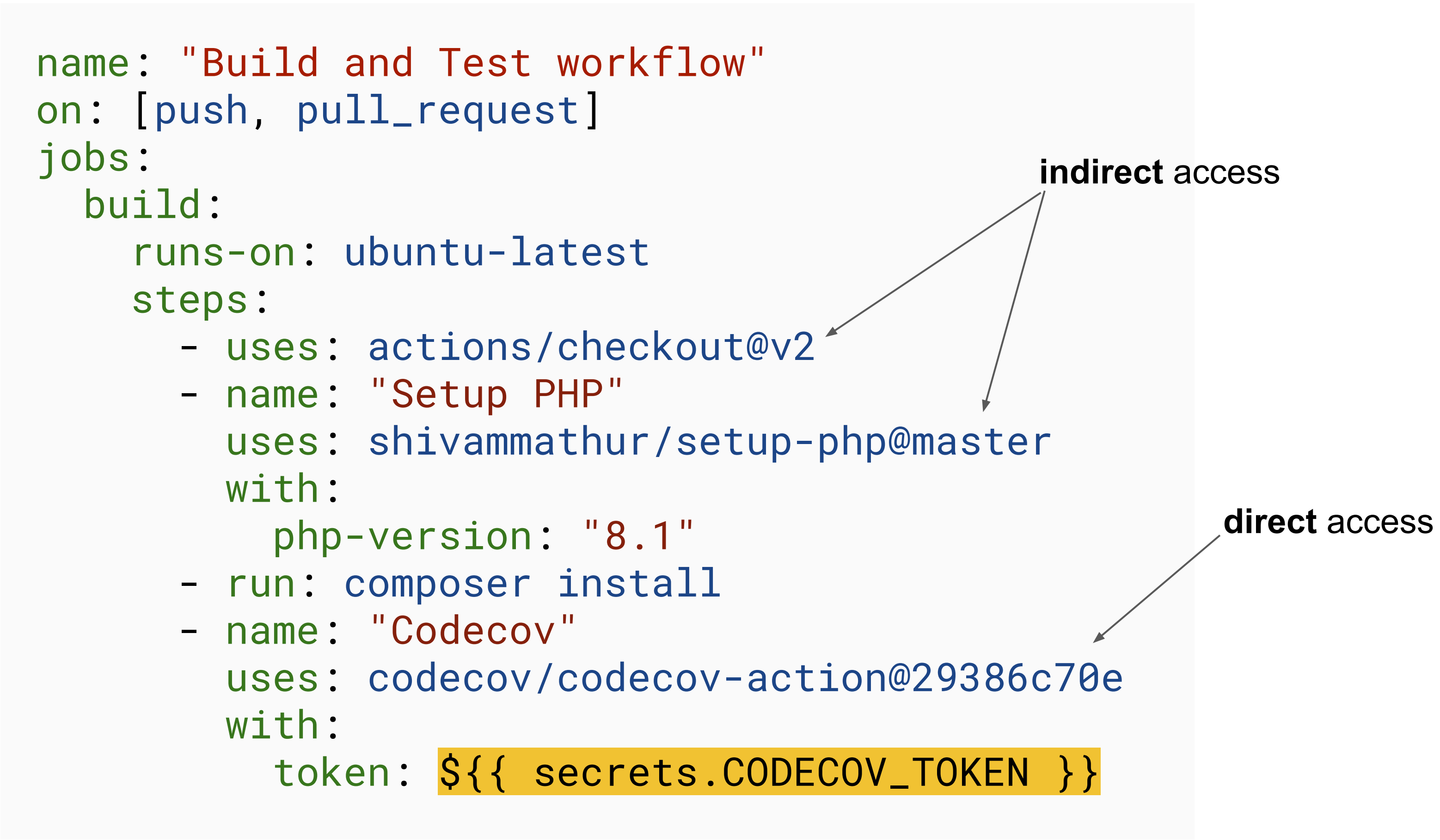
tag name

branch name

commit hash

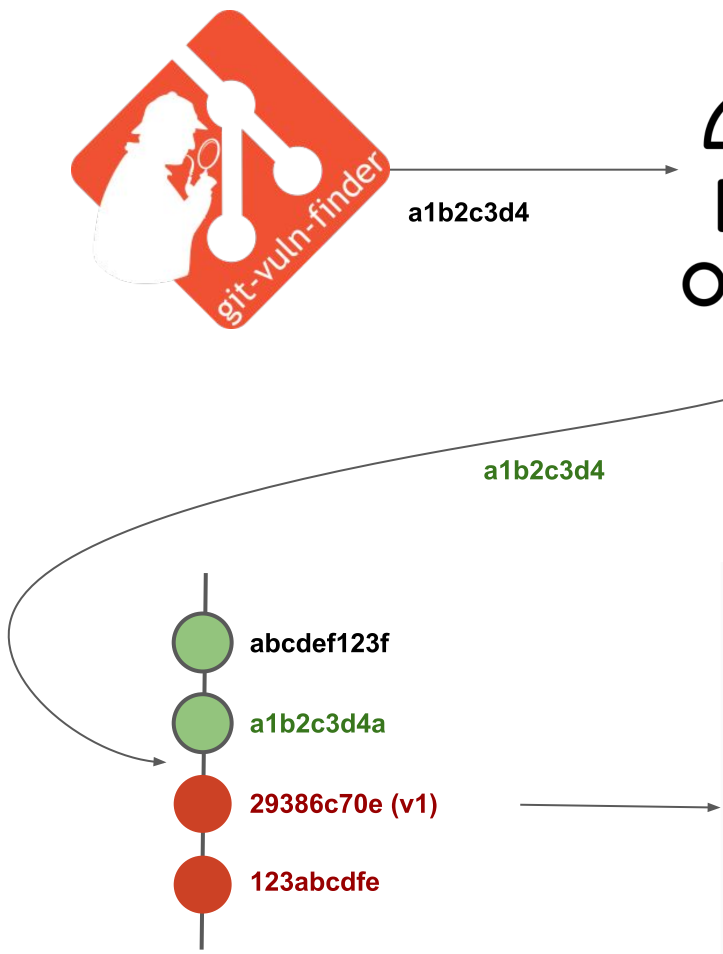
Q5: How users pass secrets to workflows?

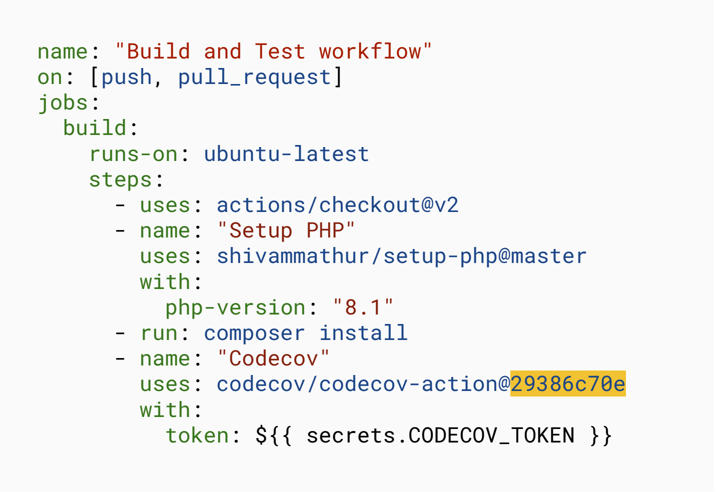
- 49.7% repositories passes the secrets
- 4,517 actions have direct access to secrets
  - only **359** (**8%**) are created by a verified creator
- 5,719 actions have *indirect* access to the secrets
  - only **53** (**0.9%**) are from verified creators



```
name: "Build and Test workflow"
on: [push, pull_request]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: "Setup PHP"
        uses: shivammathur/setup-php@master
        with:
          php-version: "8.1"
      - run: composer install
      - name: "Codecov"
        uses: codecov/codecov-action@29386c70e
        with:
          token: ${{ secrets.CODECOV_TOKEN }}
```

Q6: Do workflows depend on vulnerable plugins?





```
name: "Build and Test workflow"
on: [push, pull_request]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: "Setup PHP"
        uses: shivammathur/setup-php@master
        with:
          php-version: "8.1"
      - run: composer install
      - name: "Codecov"
        uses: codecov/codecov-action@29386c70e
        with:
          token: ${{ secrets.CODECOV_TOKEN }}
```

Vulnerability severity	Actions	Repositories
High-severity	26	582
Medium-severity	56	28,870
Low-severity	577	10,922

Vulnerable 1st and 3rd-party actions count and number of repositories that reference vulnerable versions of actions

Conclusion

- Defined four security properties that must held in CI/CD pipeline, and compared five popular CI/CD platforms
- Performed the measurement study of GitHub Workflows, and found that developers do not follow security guidelines created by GitHub
  - Only **0.2%** of repos update default permissions
  - **292** repos with `pull_request` triggered workflows that run in self-hosted machines
  - **99.9%** of third-party action references are **mutable**
  - **582** repos that reference action’s versions with high-severity vulnerability

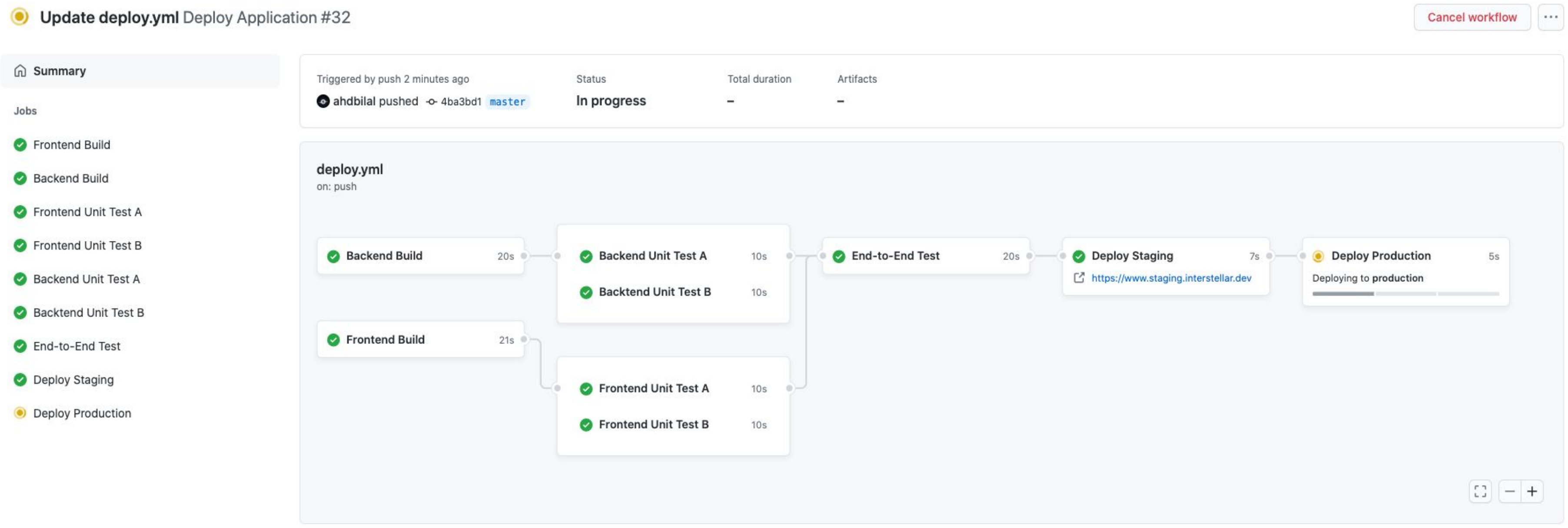
Takeaways

- CI/CD become highly dependent on third-party plugins, which makes them susceptible to supply-chain security
- Despite security guidelines, developers do NOT follow the guidelines. Therefore, platforms might need to have secure default settings, instead of trusting users to use it securely
- CI/CD platforms require more research from security professionals

Website:
https://kapravelos.com/projects/githubactions
Repository:
https://github.com/wspr-ncsu/github-actions-security-analysis

What is GitHub Actions?

1. Introduced by GitHub in 2019
2. Directly integrated into GitHub:
   a. allows developers to automate development process without leaving GitHub
   b. gaining tremendous popularity in usage
   c. need to create config file under **.github/workflows** directory
3. Features:
   a. Supports community developed plugins, called Actions
   b. Has built in Secret stores
   c. Enables to use self-hosted servers



Update deploy.yml Deploy Application #32

Triggered by push 2 minutes ago
ahdbilal pushed 4ba3bd1 master

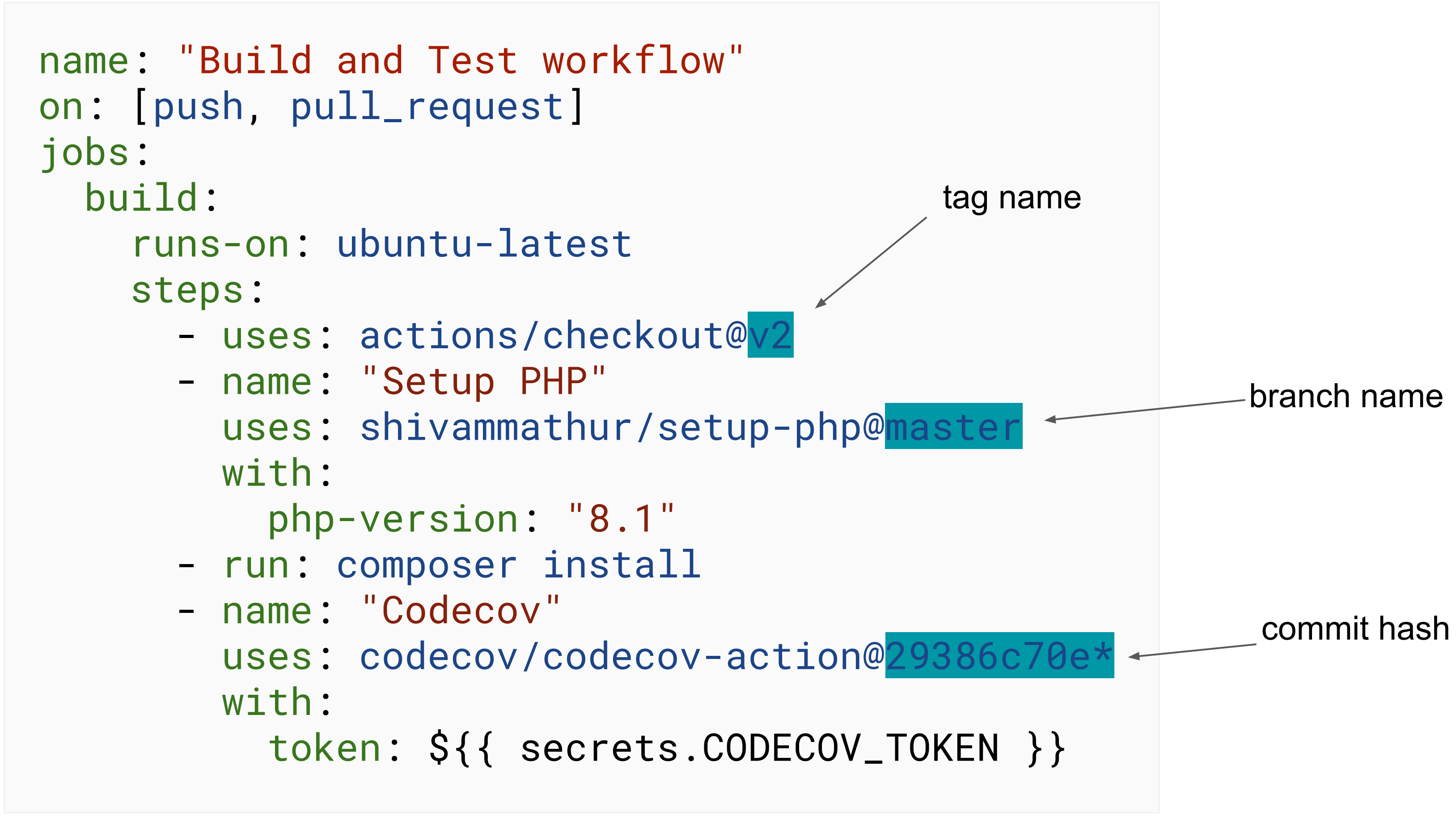
Status: In progress
Total duration: –
Artifacts: –

Jobs
- Frontend Build
- Backend Build
- Frontend Unit Test A
- Frontend Unit Test B
- Backend Unit Test A
- Backtend Unit Test B
- End-to-End Test
- Deploy Staging
- Deploy Production

deploy.yml
on: push

Third-party Actions

1. There are main three types:
   a. JavaScript
   b. Docker
   c. Composite
2. Referenced in three ways:
   a. tag (*v2*)
   b. branch (*master*)
   c. commit hash (*29386c70e**)
3. Developed by
   a. verified creator
   b. unverified creators



```yaml
name: "Build and Test workflow"
on: [push, pull_request]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: "Setup PHP"
        uses: shivammathur/setup-php@master
        with:
          php-version: "8.1"
      - run: composer install
      - name: "Codecov"
        uses: codecov/codecov-action@29386c70e*
        with:
          token: ${{ secrets.CODECOV_TOKEN }}
```

GitHub Actions VS Other CI/CD platforms

1. We compared GitHub Actions with other four popular CI/CD platforms:
   a. **TravisCI -** one of the first public (aka cloud) CI/CD platform
   b. **CircleCI -** similar to TravisCI provides servers to execute the pipeline. Supports plugin system similar to GitHub Actions
   c. **Jenkins -** the first CI/CD platform. Does not provide servers to execute.
   d. **GitLab CI -** similar to GitHub Actions in a sense it also provides VCS