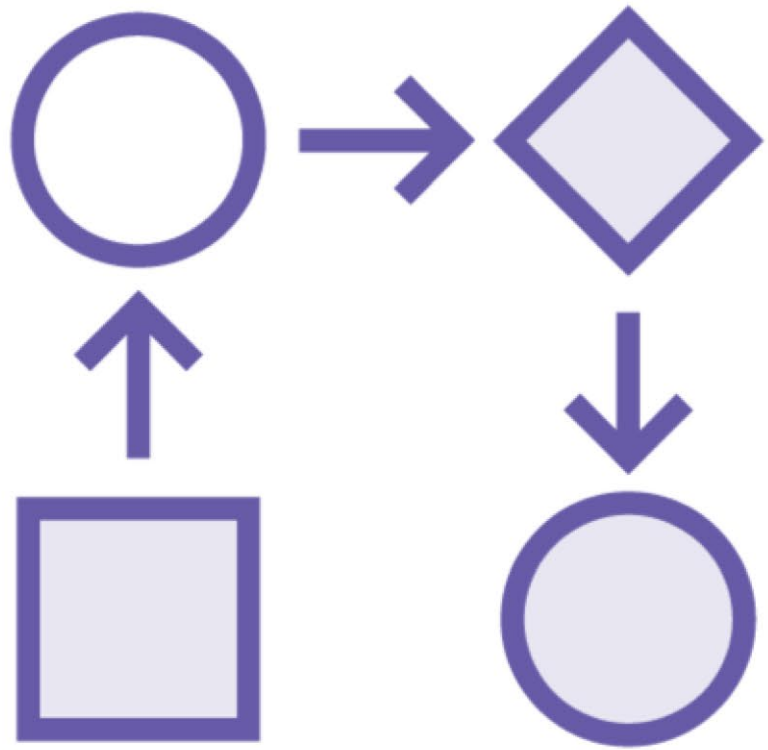
Getting Started with GitHub Actions

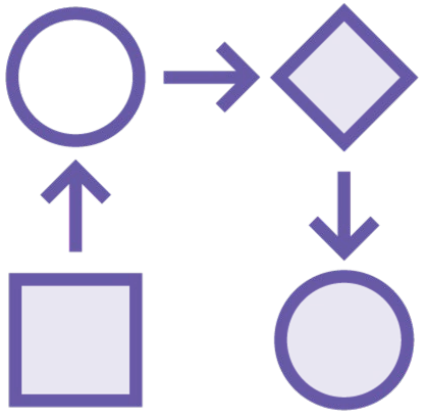
CORE CONCEPTS OF A GITHUB ACTION

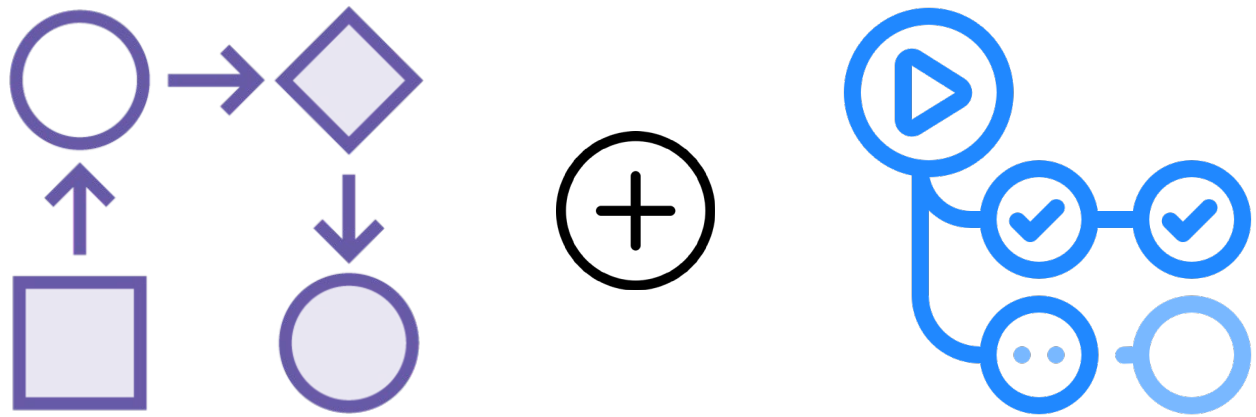
Aaron Stewart
PROGRAM ARCHITECT

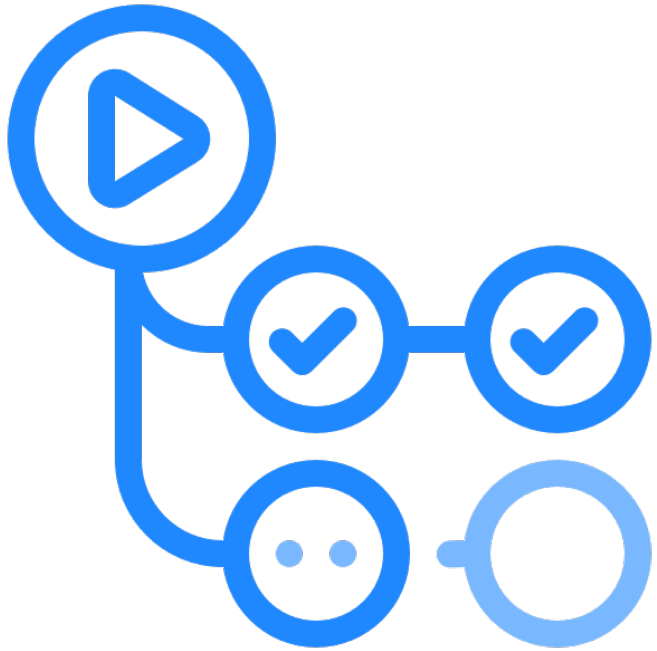
www.github.com/a-a-ron



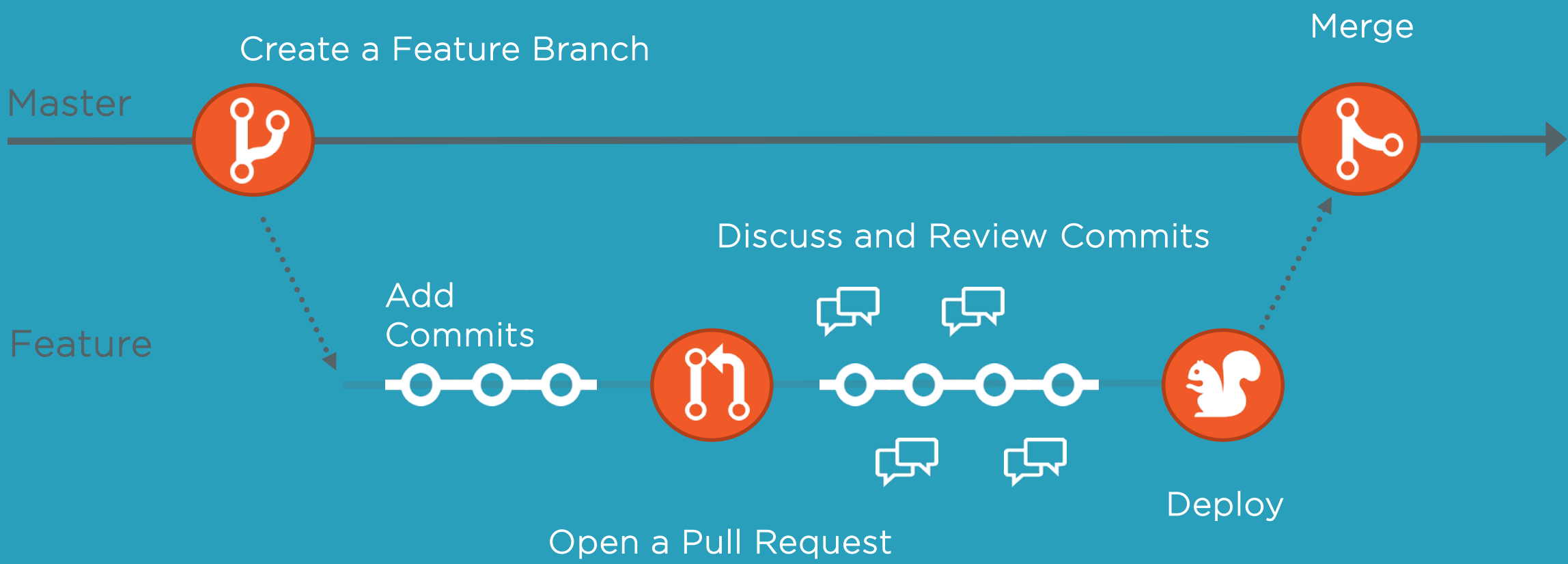

STAR







GitHub Flow


Create a Feature Branch
Merge
Master
Discuss and Review Commits
Add
Commits
Feature
Deploy
Open a Pull Request

Supercharged GitHub Flow

Require an approving review before it can be merged

Supercharged GitHub Flow

GitHub Actions in Action

```yaml
on:
  push:
    branches:
      - master
    tags:
      - v1
    paths:
      - 'test/*'
```

- Push event
- Only for the master branch
- Only commits pushed to the v1 tag
- Only files that are in the test directory

GitHub-Hosted Runners

Virtual environment	YAML workflow label
Windows Server 2019	`windows-latest` or `windows-2019`
Ubuntu 18.04	`ubuntu-latest` or `Ubuntu-18.04`
Ubuntu 16.04	`ubuntu-16.04`
macOS Catalina 10.15	`macos-latest` or `macos-10.15`

Self-Hosted Runners

Linux

- Red Hat Enterprise Linux 7
- CentOS 7
- Oracle Linux 7
- Fedora 29 or later
- Debian 9 or later
- Ubuntu 16.04 or later
- Linux Mint 18 or later
- openSUSE 15 or later
- SUSE Enterprise Linux (SLES) 12 SP2 or later

MacOS

- macOS 10.13 (High Sierra) or later

Windows

- Windows 7 64-bit
- Windows 8.1 64-bit
- Windows 10 64-bit
- Windows Server 2012 R2 64-bit
- Windows Server 2016 64-bit
- Windows Server 2019 64-bit

Self-Hosted Runners

```
runs-on: [self-hosted, linux, ARM32]
```

- `self-hosted` - **Run this job on a self-hosted runner**
- `linux` - **Only use a Linux-based runner**
- `ARM32` - **Only use a runner based on ARM64 hardware**

https://help.github.com/en/actions/hosting-your-own-runners/about-self-hosted-runners

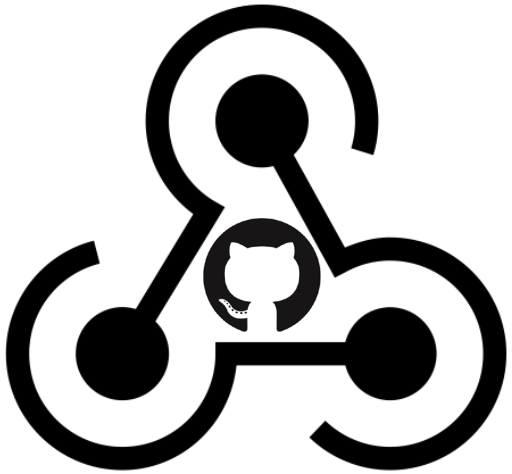
GitHub Events

`on: pull_request_comment` `on: delete` `on: release` `on: deployment` `on: pro`

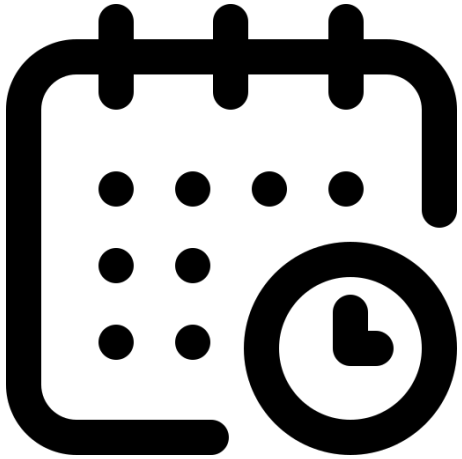
`on: check_suite` `on: pull_request` `on: pull_request_review` `on: scheduled`

`on: issue_comment` `on: push` `on: page_build` `on: check_run` `on: mile`

GitHub Events

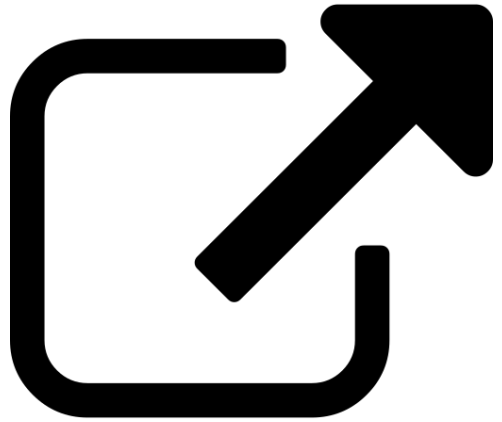


GitHub Webhook Events



Scheduled Events

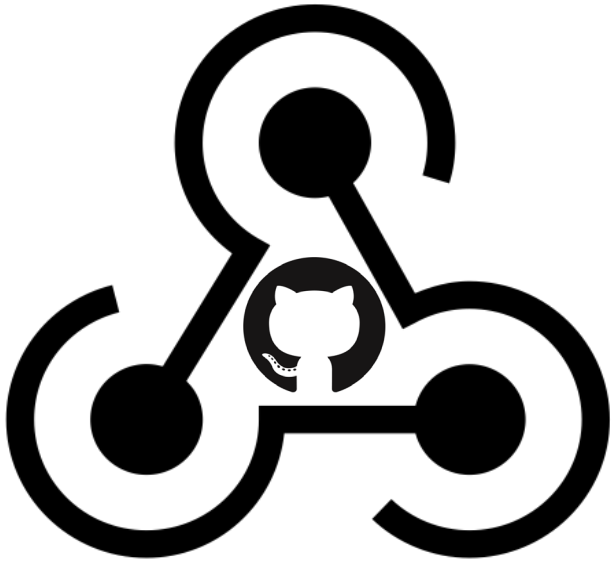
`schedule`



External Events

`Repository_dispatch`

GitHub Events



You can configure your workflow to run when webhook events are created on GitHub

```
on: [push, pull_request]
```

GitHub Events

Webhook Event Payload	Activity Types
pull_request	- assigned - unassigned - review_requested - review_request_removed - labeled - unlabeled - Opened - edited - closed - ready_for_review - locked - unlocked - reopened

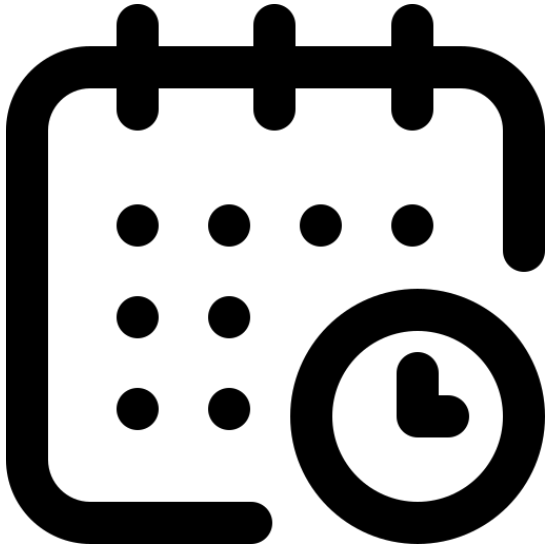
GitHub Events

```
on:
    pull_request:
        types: [assigned, unassigned]
```

GitHub Events

```
on:
    issue_comment:
        types: [created, deleted]
```

GitHub Events

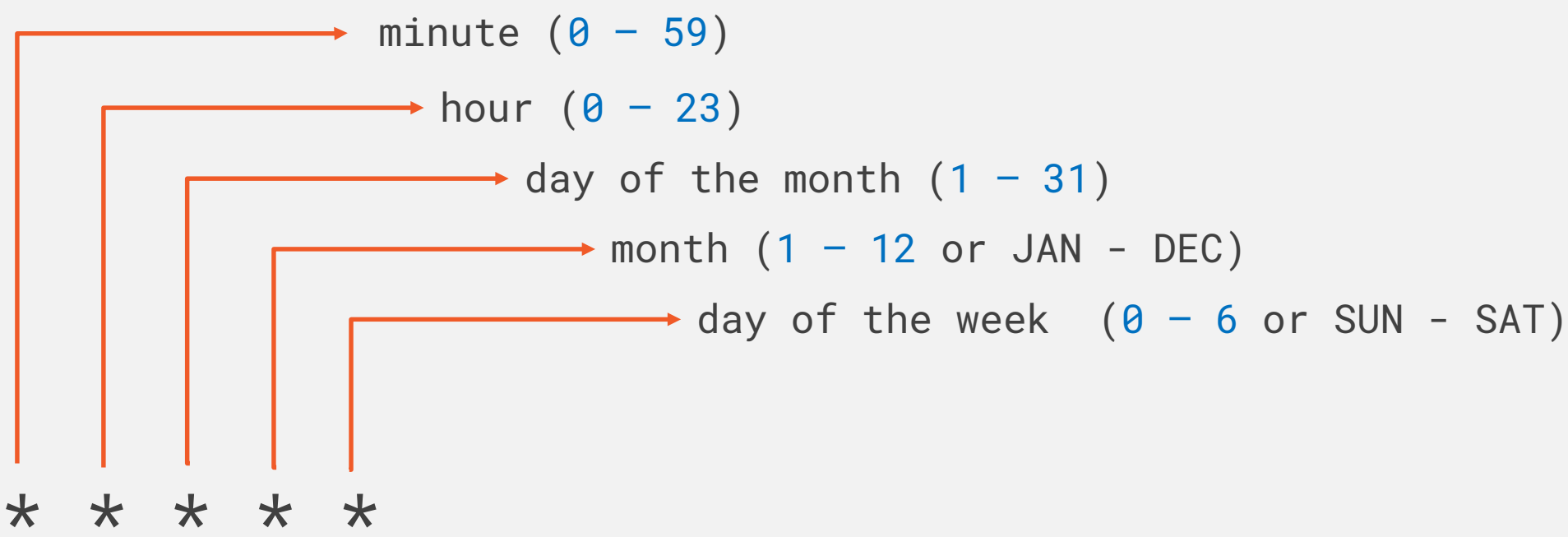


You can schedule a workflow to run at specific UTC times using POSIX cron syntax.

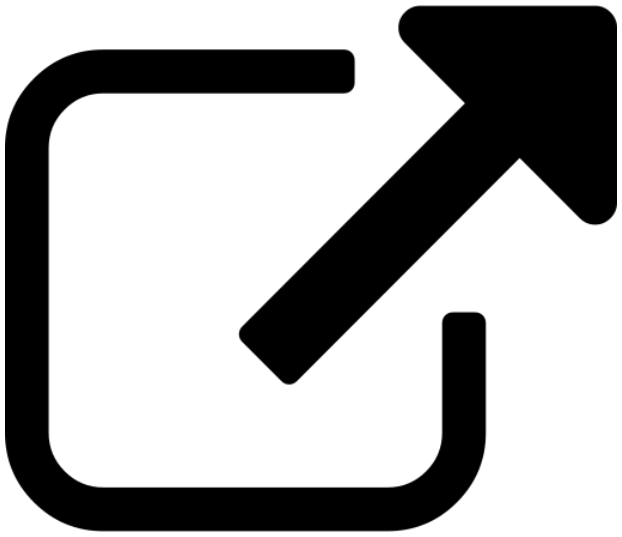
Run on the latest commit on the default or base branch

```
on:
  schedule:
    - cron: '*/15 * * * *'
```

GitHub Events



GitHub Events

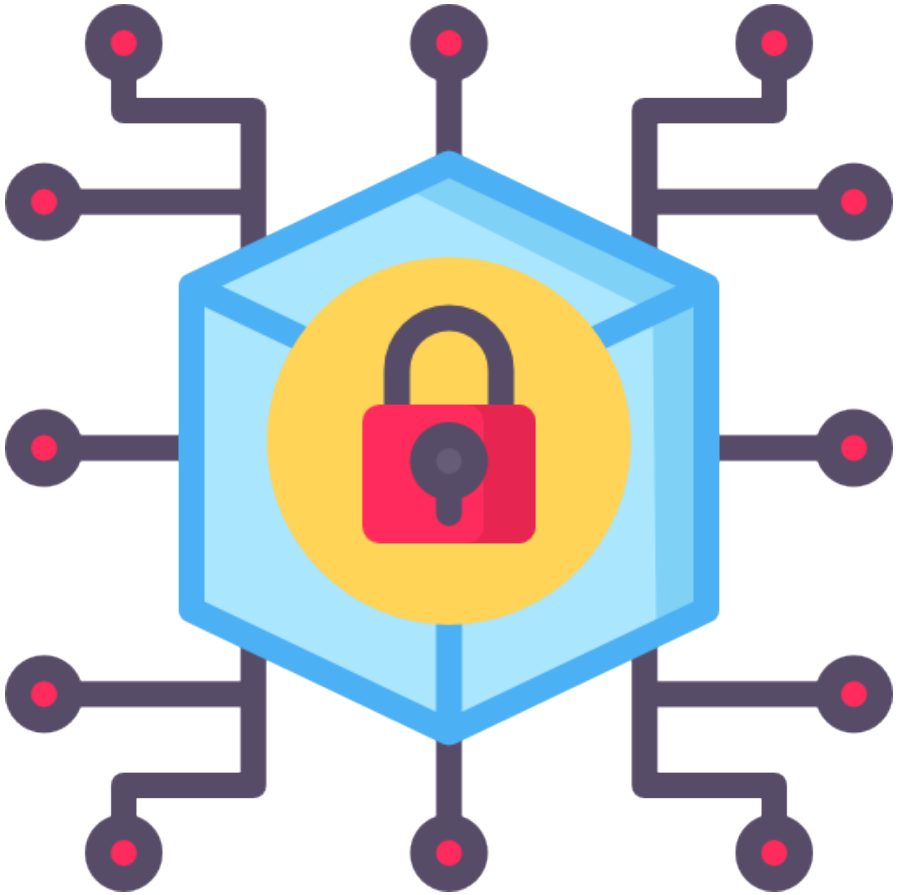


Send a POST request to the GitHub API endpoint

Inlcude `repository_dispatch` with identified types in your workflow file

```
on: repository_dispatch
    types: [opened, deleted]
```

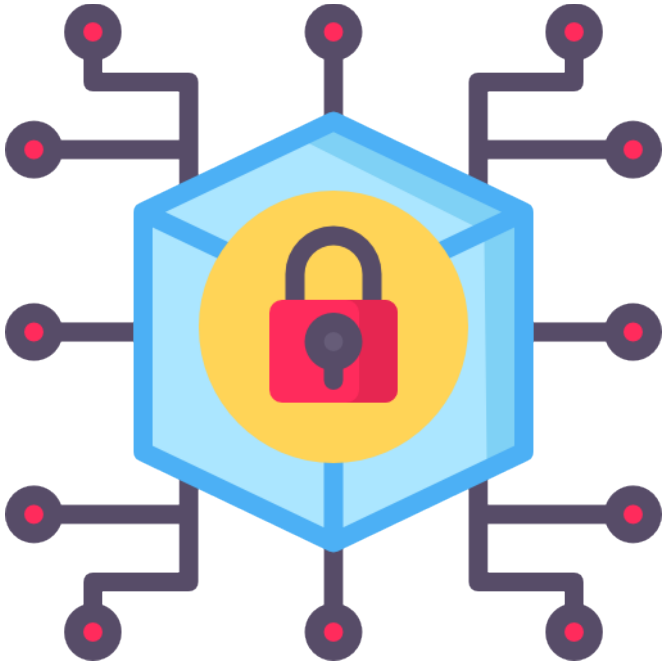
Environment Variables and Secrets



Environment Variables and Secrets

```
steps:
  - name: Hello world
    with:
      super_secret: ${{ secrets.MySecret }}
    env:
      super_secret: ${{ secrets.MySecret }}
```

Environment Variables and Secrets



`GITHUB_TOKEN`

An access token that you can use to authenticate on behalf of the GitHub App installed on your repository

Need to reference it in your workflow file

Environment Variables and Secrets

```
name: Create issue on commit
on:
-  push
jobs:
  create_commit:
    runs-on: ubuntu-latest
    steps:
    - name: Create issue using REST API
      run: |
        curl --request POST \
        --url https://api.github.com/repos/${{ github.repository }}/issues \
        --header 'authorization: Bearer ${{ secrets.GITHUB_TOKEN }}' \
        --header 'content-type: application/json' \
        --data '{
          "title": "Automated issue for commit: ${{ github.sha }}",
          "body": "This issue was automatically created by the GitHub Action
                   workflow **${{ github.workflow }}**. \n\n The commit hash was:
                   _${{ github.sha }}_."
          }'
```

Core Concepts Overview

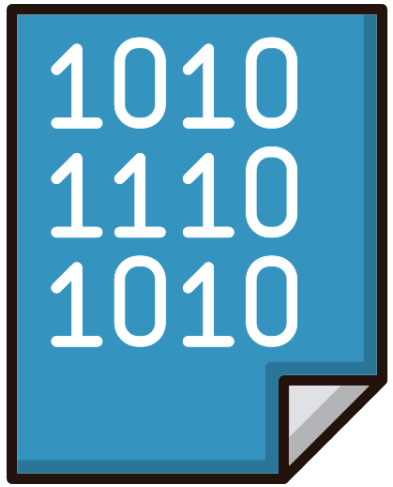
Core Concepts Overview

- Actions are your reusable units of code.
- You can use Action from the marketplace or create one on your own
- To use an action in a workflow run, you must include it as a step

Core Concepts Overview

Actions

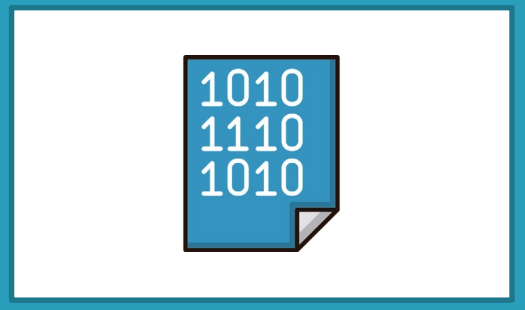
Core Concepts Overview



- Artifacts are the files created when you build and test your code
- Binary, package, or log files

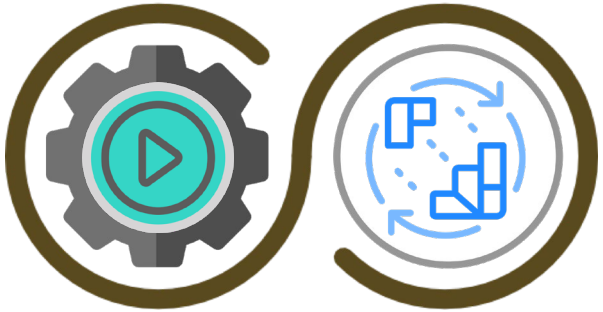
Core Concepts Overview

Actions



Artifacts

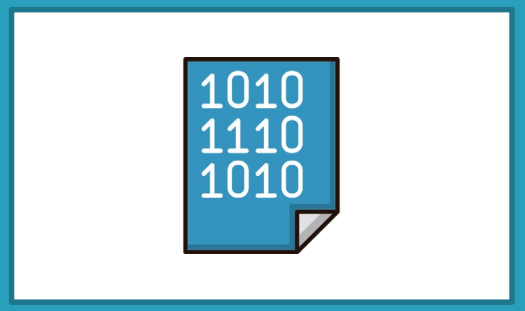
Core Concepts Overview



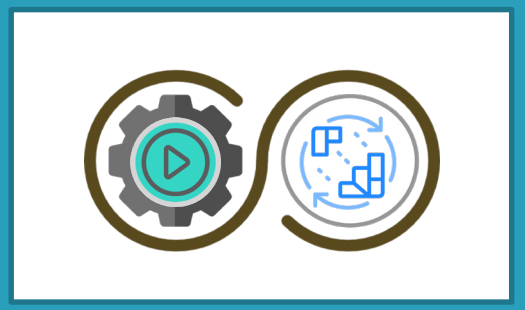
- Create custom CI workflows that automate building and testing of our code
- Create custom CD workflows to automatically deploy our code to any cloud, self-hosted service, or platform from our repository

Core Concepts Overview

Actions



Artifacts



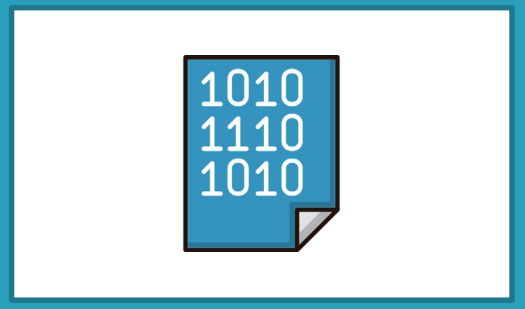
CI/CD

Core Concepts Overview

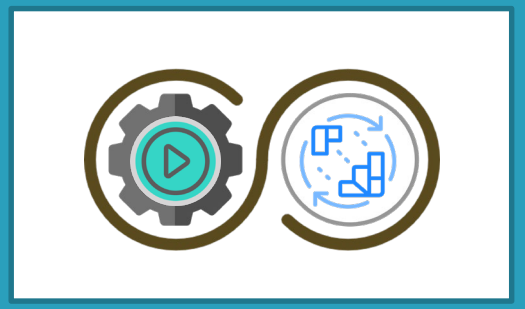
- **Specific activities that trigger a workflow run**
- on :
- **Provide a single event, an array of events, or an array of event activity types**

Core Concepts Overview

Actions



Artifacts



CI/CD

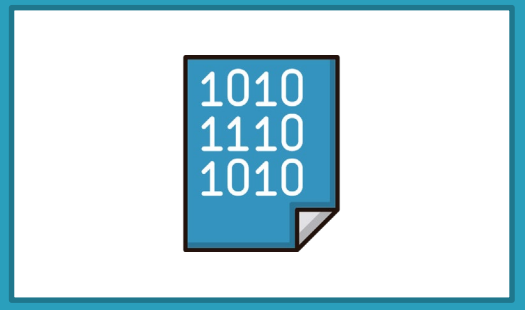
Events

Core Concepts Overview

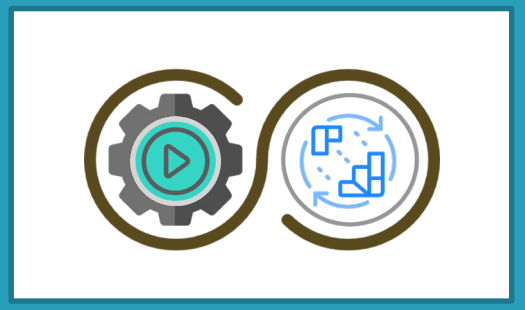
- This tells our workflow run where the virtual environment will be
- Wait for an available job to be kicked off when an event is triggered
- Only run one job at a time

Core Concepts Overview

Actions

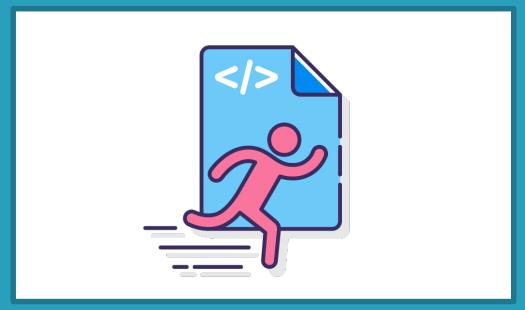


Artifacts



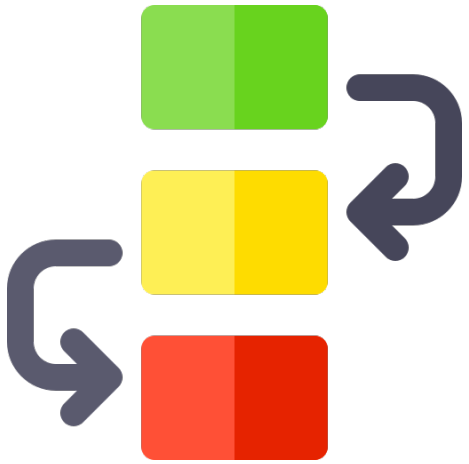
CI/CD

Events



Runners

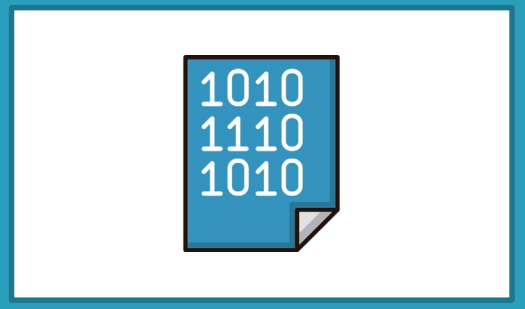
Core Concepts Overview



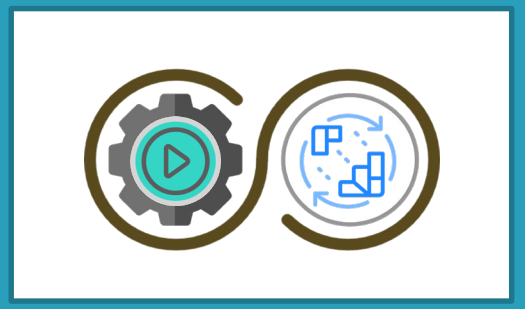
- A job consists of one or more steps
- Jobs can run independently or sequentially
- A step is a set of tasks that can be executed by a job

Core Concepts Overview

Actions

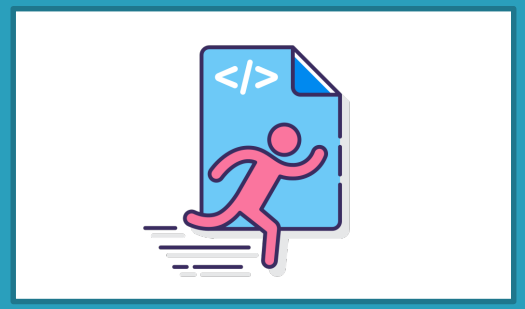


Artifacts

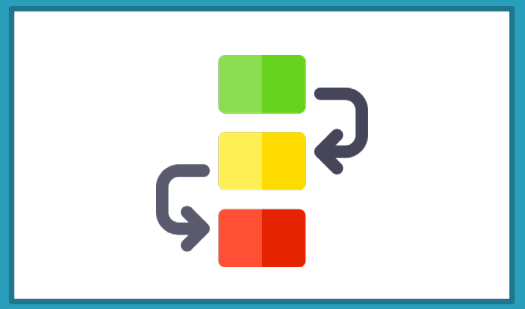


CI/CD

Events

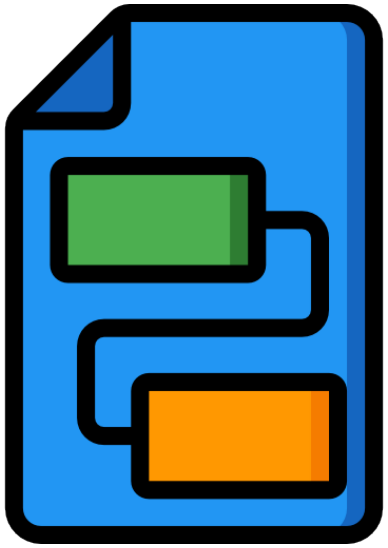


Runners



Jobs/Steps

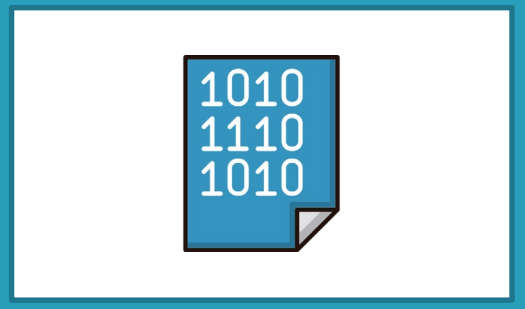
Core Concepts Overview



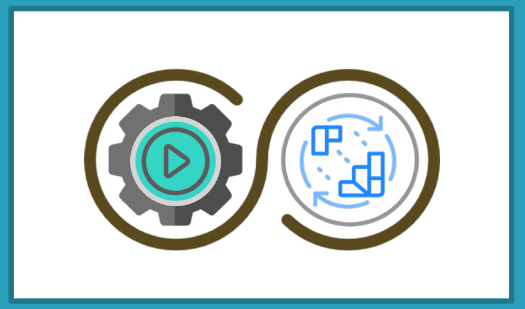
- An automated process that is made up of one or multiple jobs and can be triggered by an event
- Defined using a YAML file in the `.github/workflows` directory.

Core Concepts Overview

Actions

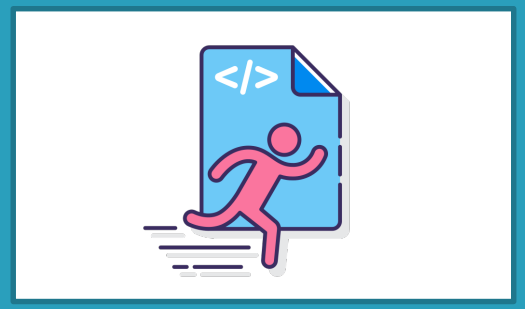


Artifacts

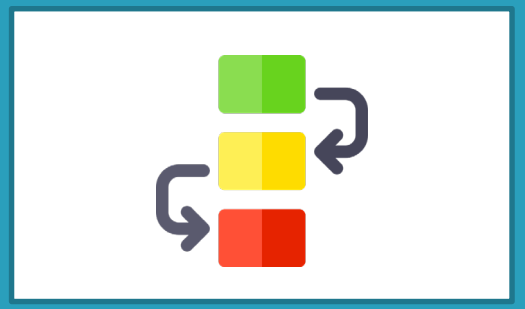


CI/CD

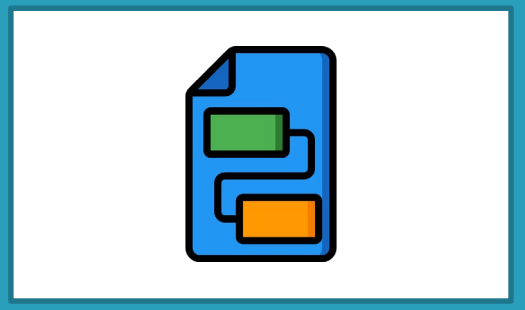
Events



Runners

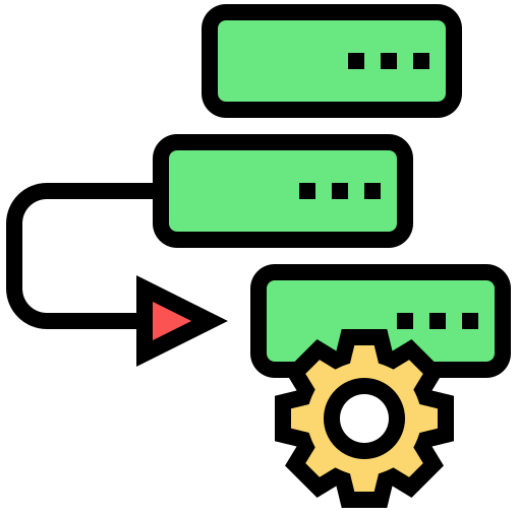


Jobs/Steps



Workflow File

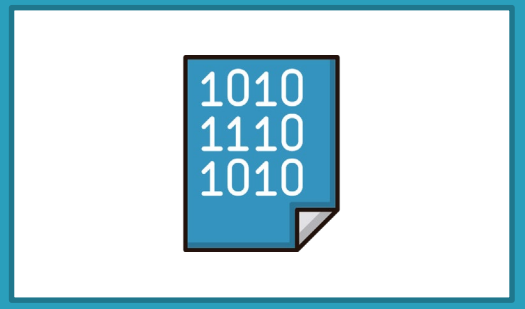
Core Concepts Overview



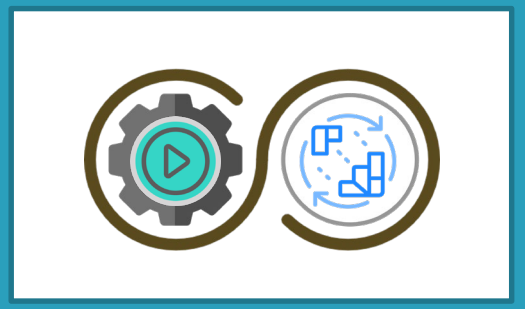
- instance of your workflow that runs when the pre-configured event occurs
- You can see the jobs, actions, logs, and statuses for each workflow run

Core Concepts Overview

Actions

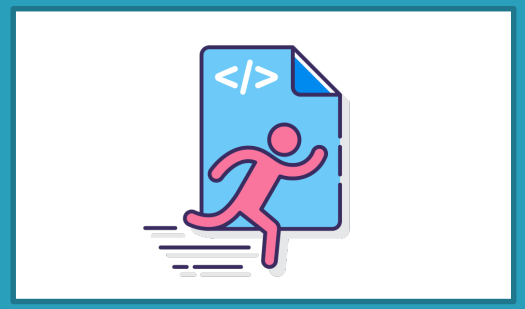


Artifacts

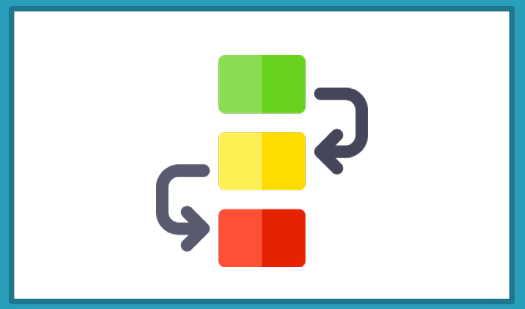


CI/CD

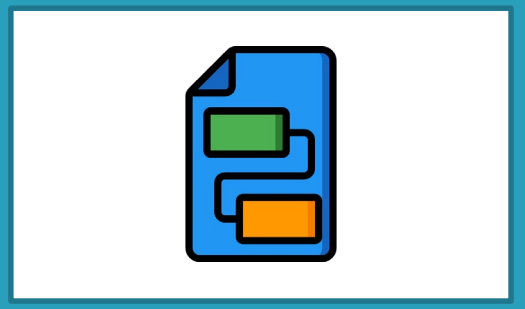
Events



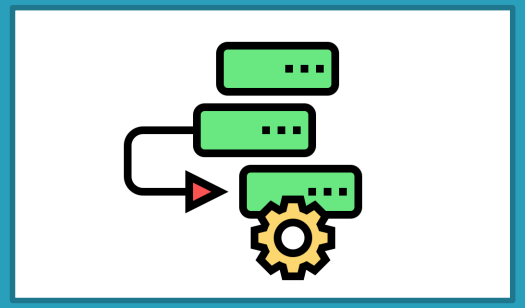
Runners



Jobs/Steps



Workflow File



Workflow Run

In Summary

Supercharged our GitHub workflow to the ultimate level with GitHub Actions

Created a simple workflow file and triggered a workflow run with some simple bash commands

Dissected the workflow file

Navigated our workflow run logs

Learned about event triggers, repository secrets and environment variables

Reviewed core concepts of GitHub Actions