

VisugXL

GitHub Code To Cloud Workshop



# GitHub

**Marcel de Vries**

**Rob Bos**

**Dennis Thie**

---



## Marcel de Vries

Founder and CTO at Xpirit  
Microsoft Regional Director, MVP  
[linkedin.com/in/marcelv](https://linkedin.com/in/marcelv)  
mdevries@xpirit.com



## Rob Bos

DevOps Consultant at Xpirit  
Microsoft MVP  
[linkedin.com/in/bosrob](https://linkedin.com/in/bosrob)  
rbos@xpirit.com



## Dennis Thie

Consultant at Xpirit  
[linkedin.com/in/dennisthie](https://linkedin.com/in/dennisthie)  
dthie@xpirit.com

Let's connect!



# GitHub Code to Cloud Workshop Agenda

	<b>Introduction to GitHub</b>	13.30 – 14.00
	Migrating a repository from Azure DevOps to GitHub	14.00 – 14.15
	Codespaces – Your development IDE in the cloud	14.15 – 14.20
	Setting up Codespaces to develop a web-app	14.20 – 14.35
	GitHub Actions	14.35 – 15.00
	Coffee break	15.00 – 15.15
	Creating your first Actions workflow	15.15 – 15.30
	Creating a .NET Actions workflow	15.30 – 16.00
	GitHub Advanced Security	16.00 – 16.15
	Code Scanning and Secret Scanning	16.15 – 16.35
	Dependabot	16.35 – 16.50
	Wrap-up	16.50 – 17.00

# Pre-requisites for the workshop



Please ensure you have Git installed on your local machine.  
You can find download and install instructions here:  
<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>



You need access to two repositories:

1) Please ensure you can access the **Workshop repository**

<https://github.com/XpiritCommunityEvents/HOL>

2) Please ensure you can access **your attendee repository**

<https://github.com/XpiritCommunityEvents/attendee-<yourGitHubhandle>>

“

**Our highest priority is to satisfy the  
customer through early and continuous  
delivery of valuable software**

-- 1<sup>st</sup> principle behind agile manifesto



# Continuous Delivery

Continuous delivery is all about creating a repeatable and reliable process for delivering software in order to **deliver high value software to our customers fast!**



“

**DevOps is the union of people, process,  
and products to enable continuous  
delivery of value to our end users.**

-- Donovan Brown



# **Awesome, but how do we do this?**

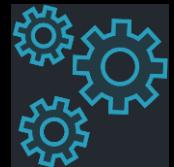
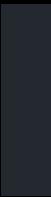
In a secure and compliant way?



# Meet the GitHub Toolset



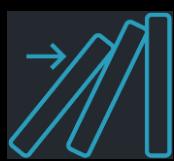
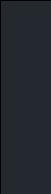
Project/Product Management



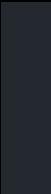
Automation



Package Management



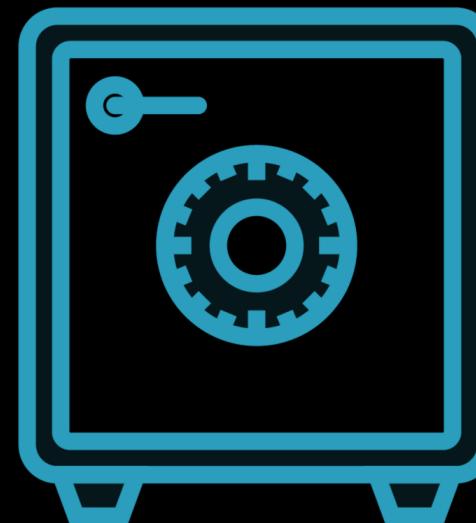
Software Supply Chain



Work From Anywhere



**Source Control**



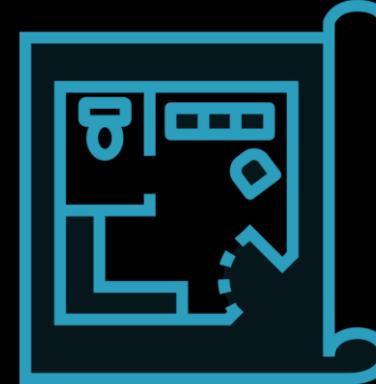
# Issues, Projects and Pages



Issues



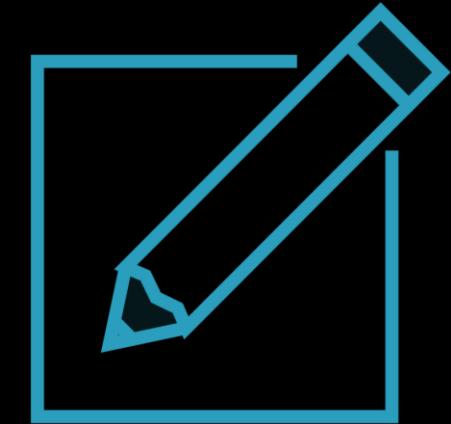
Kanban



Projects



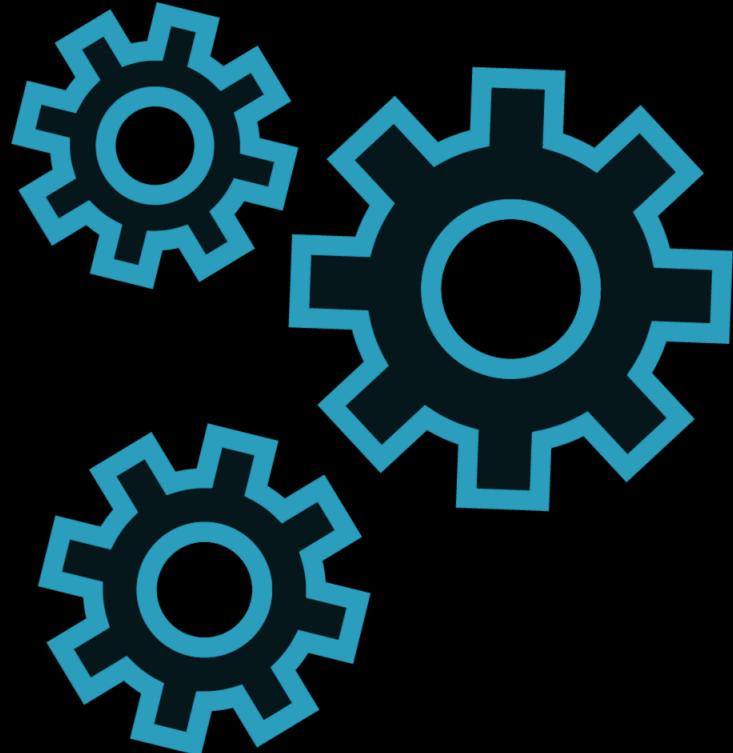
Backlog



Pages/wiki



# Automation



GitHub Actions  
Continuous integration  
Deployment automation  
Continuous delivery  
Traceability and compliance  
Hooks for other products



# GitHub Packages



Package registry  
A standard package manager  
NPM (NodeJS)  
NuGet (.NET)  
RubyGems (Ruby)  
Maven and Gradle (Java)  
Container Registry  
Unified Identity and permissions



# Software Supply Chain



Your software is build on other software  
Has dependencies  
Has known vulnerabilities  
Keep it secure  
Keep it up to date  
Scan for known vulnerabilities



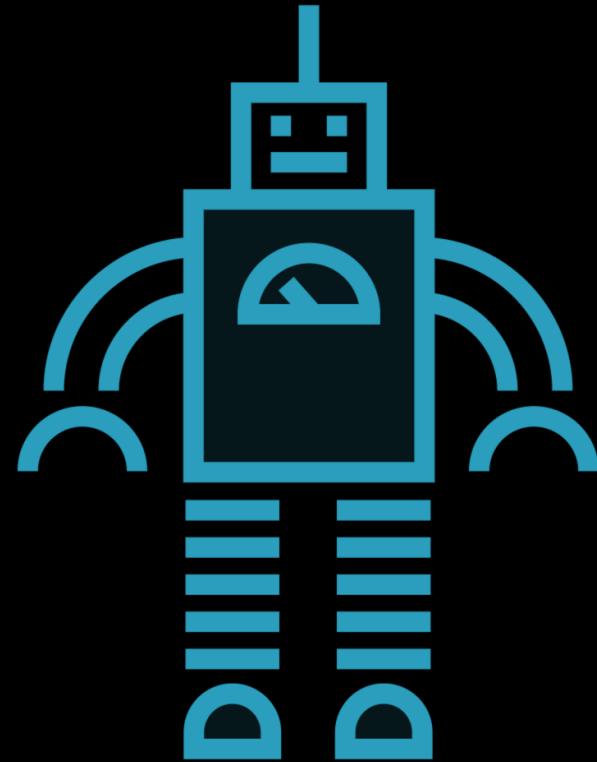
# GitHub Advanced Security



Works With GitHub Enterprise  
Code Scanning  
CodeQL  
SARIF for 3rd party tools  
Secret Scanning  
Many secrets of known service providers



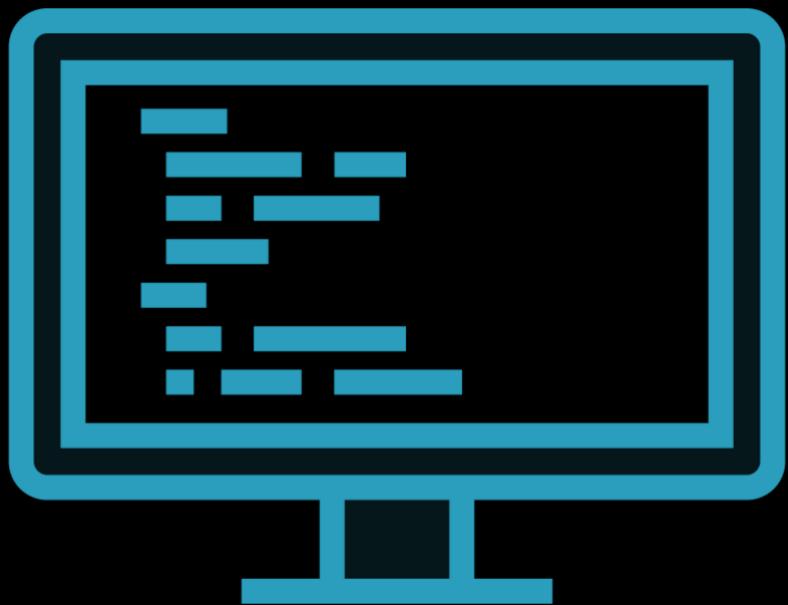
# Dependabot



GitHub tooling to keep dependencies up to date  
Outdated Packages  
Vulnerable Packages  
Actions in your workflows



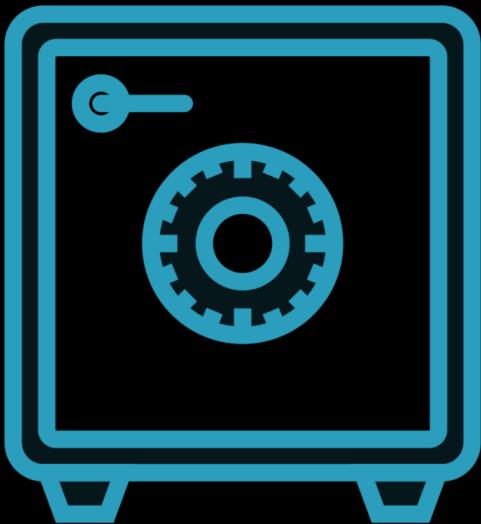
# Codespaces



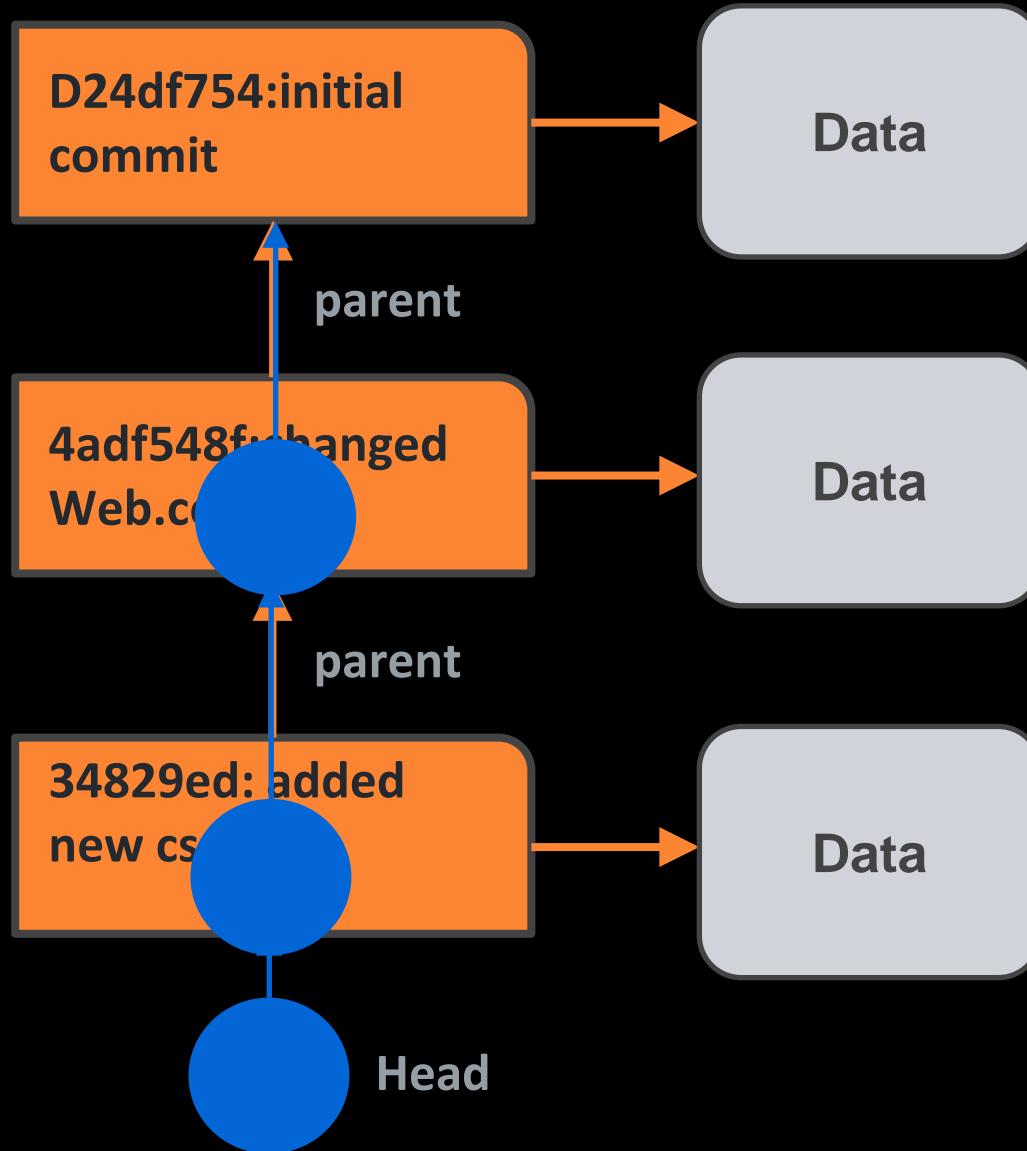
Development environment hosted in the cloud  
Different VM sizes  
Browser or Visual Studio Code  
`devcontainer.json`  
Custom container



# SOURCE CONTROL



# Git Is All About a Graph of Nodes!



A Git commit is a node in a graph

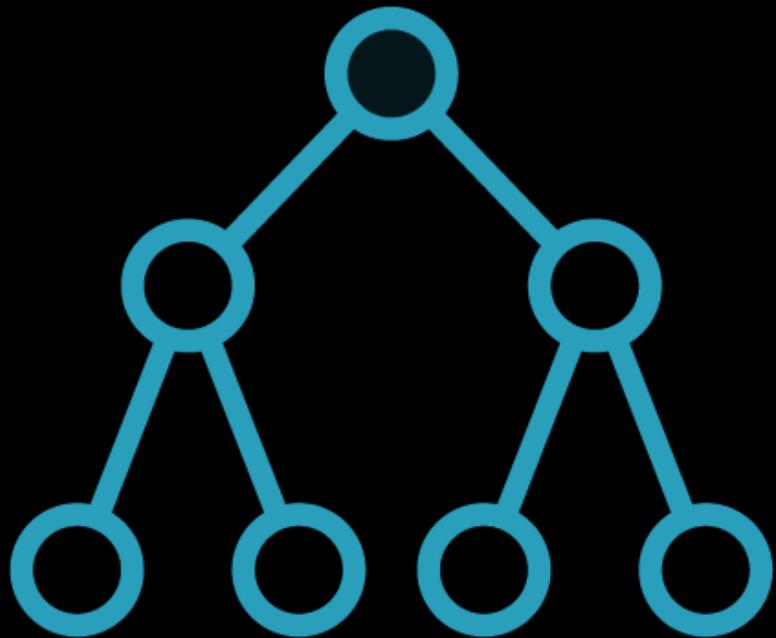
Every commit has a pointer to its parent

References make commits reachable

Head, Tag, Branch



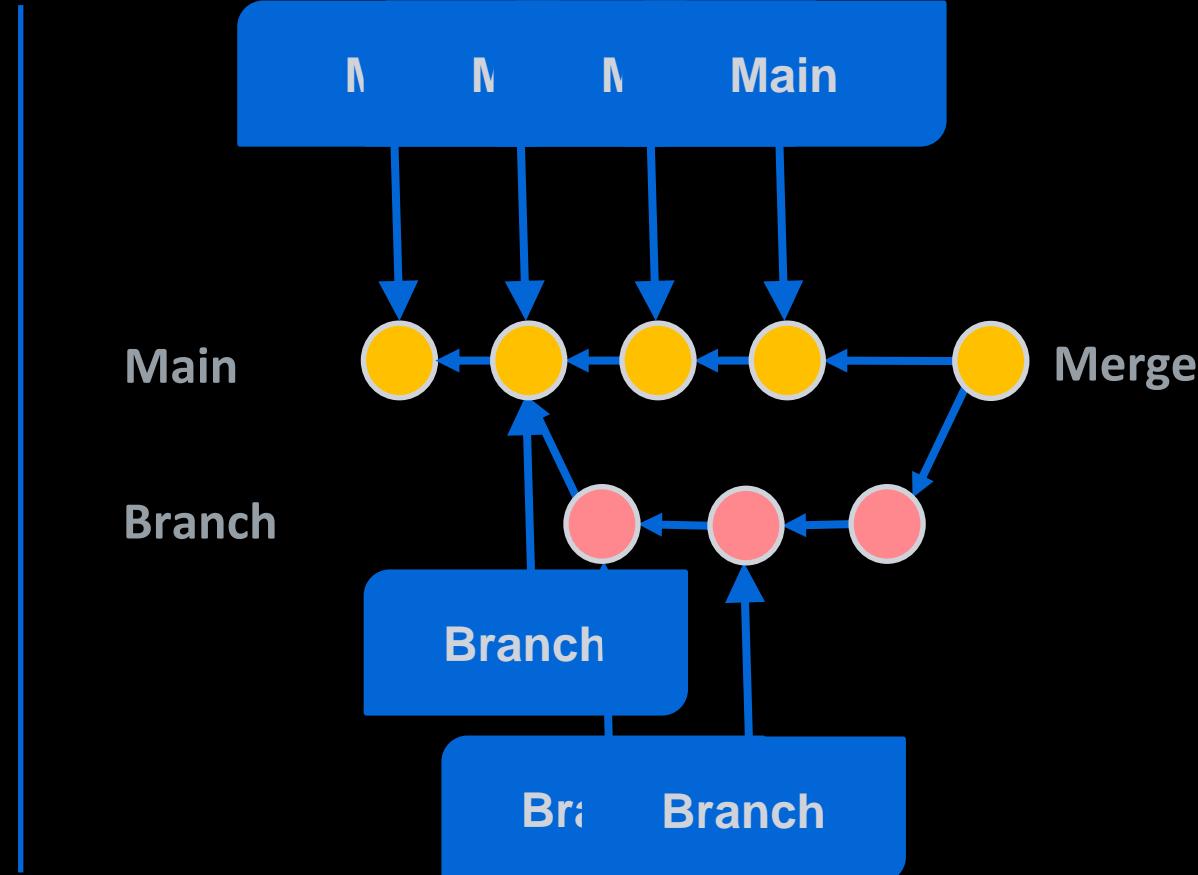
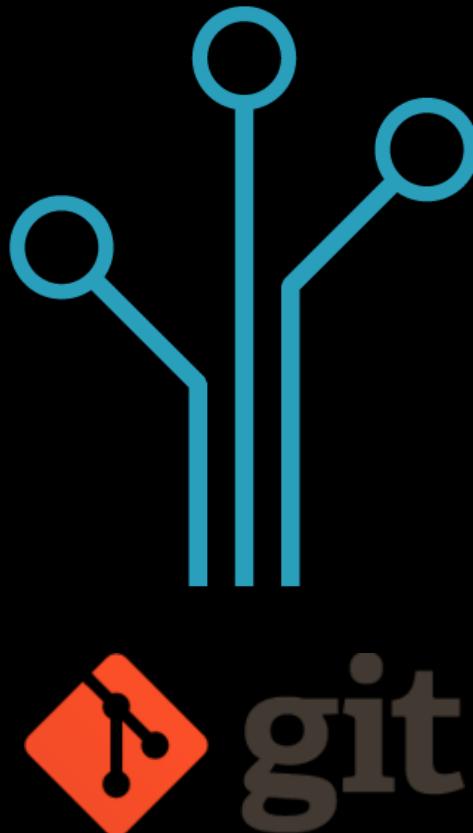
# Branches in Git



Main  
Branch + Publish  
Merge



# Branch and Merge



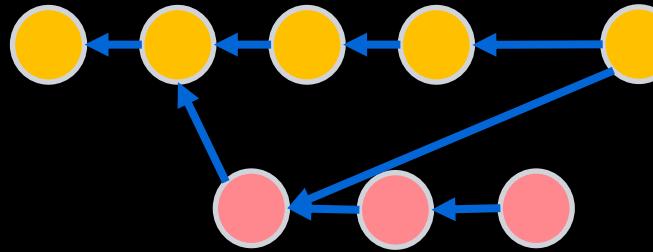
# Merge and Rebase



Main

branch

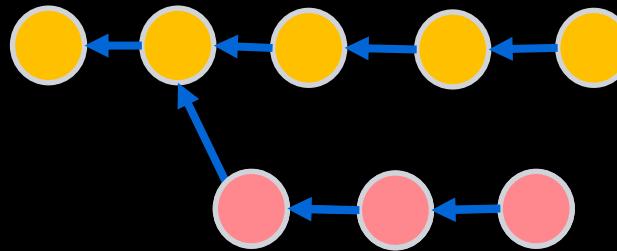
Squash & Merge



Main

branch

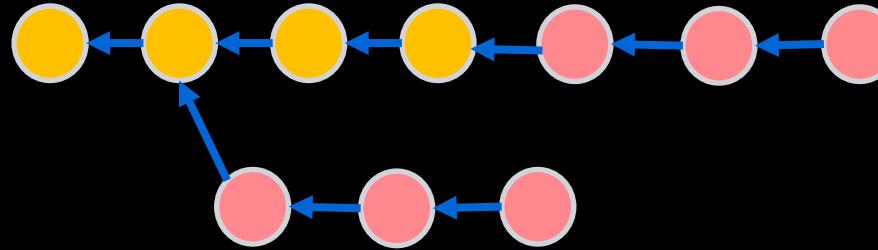
Fast forward Merge



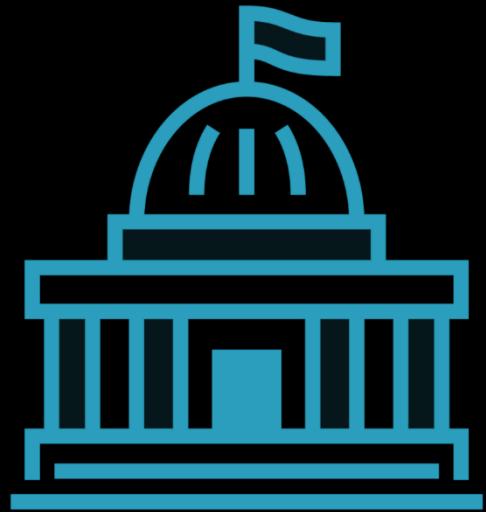
Main

branch

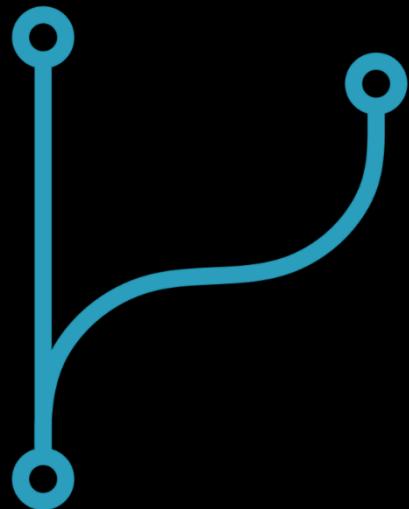
Rebase



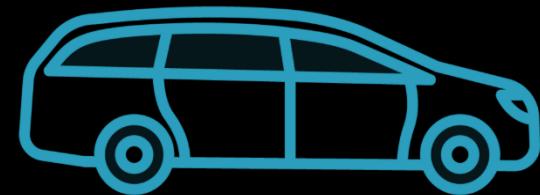
# Most Common Git Branching Strategies



**Git flow**  
Low deployment frequency



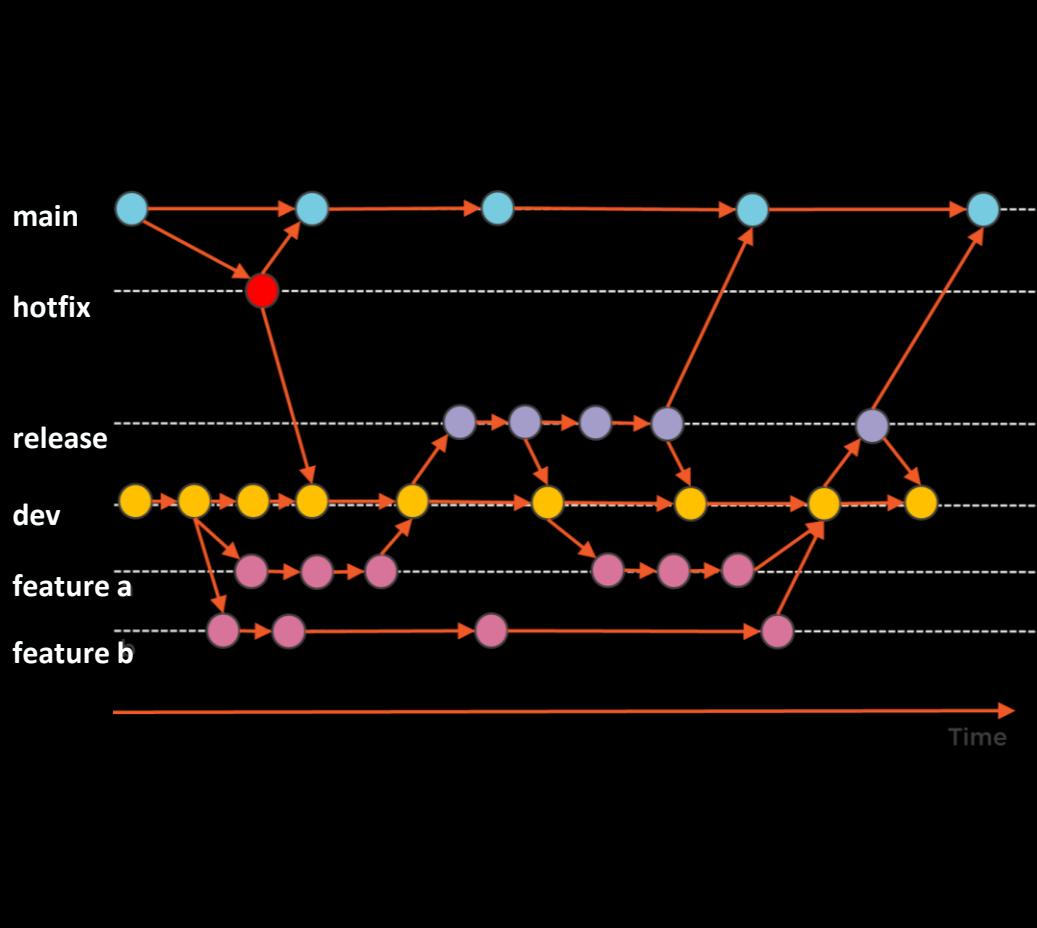
**Git Hub Flow**  
High deployment frequency



**Trunk Based Development**  
High Deployment frequency



# Git Flow



Used for staged delivery  
Release every iteration  
Delivery of packages, libraries,  
etc.

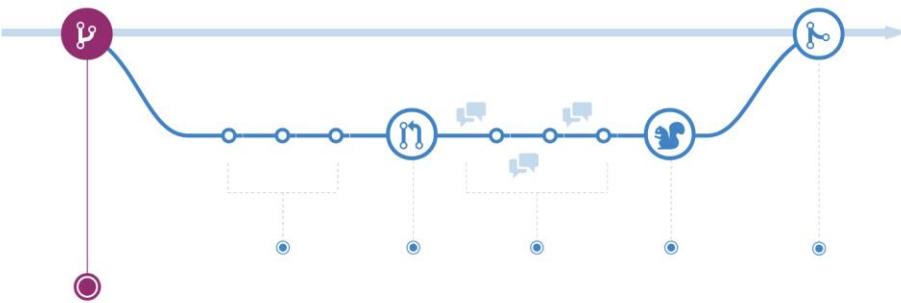
# A GitHub flow guide

[GitHub Guides](#)      [Video Guides](#) [GitHub Help](#) [GitHub.com](#)

## Understanding the GitHub flow

⌚ 5 minute read [Download PDF version](#)

GitHub flow is a lightweight, branch-based workflow that supports teams and projects where deployments are made regularly. This guide explains how and why GitHub flow works.



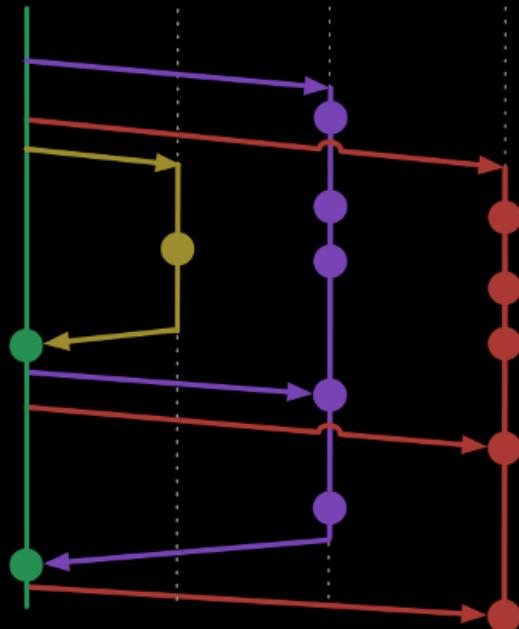
**Create a branch**

When you're working on a project, you're going to have a bunch of different features or ideas in progress at any given time – some of which are ready to go, and others which are not. Branching exists to help you

<https://guides.github.com/introduction/flow>



# Trunk Based Development



Almost the same as GitHub flow  
Release always from main  
Merge to main before you deploy  
Use feature toggles to carry cross multiple integrations to main



# Experiment without risk

## Branch:

- Lightweight pointer
- Safe to experiment
- New commits
- 1 new branch to 1 Pull Request

## Pull Request:

- Compare two branches
- Automation
  - CI
  - Deployment
- Fast feedback **in context**
  - Automation results
  - Peer reviews



# Branch vs Fork

## Branch:

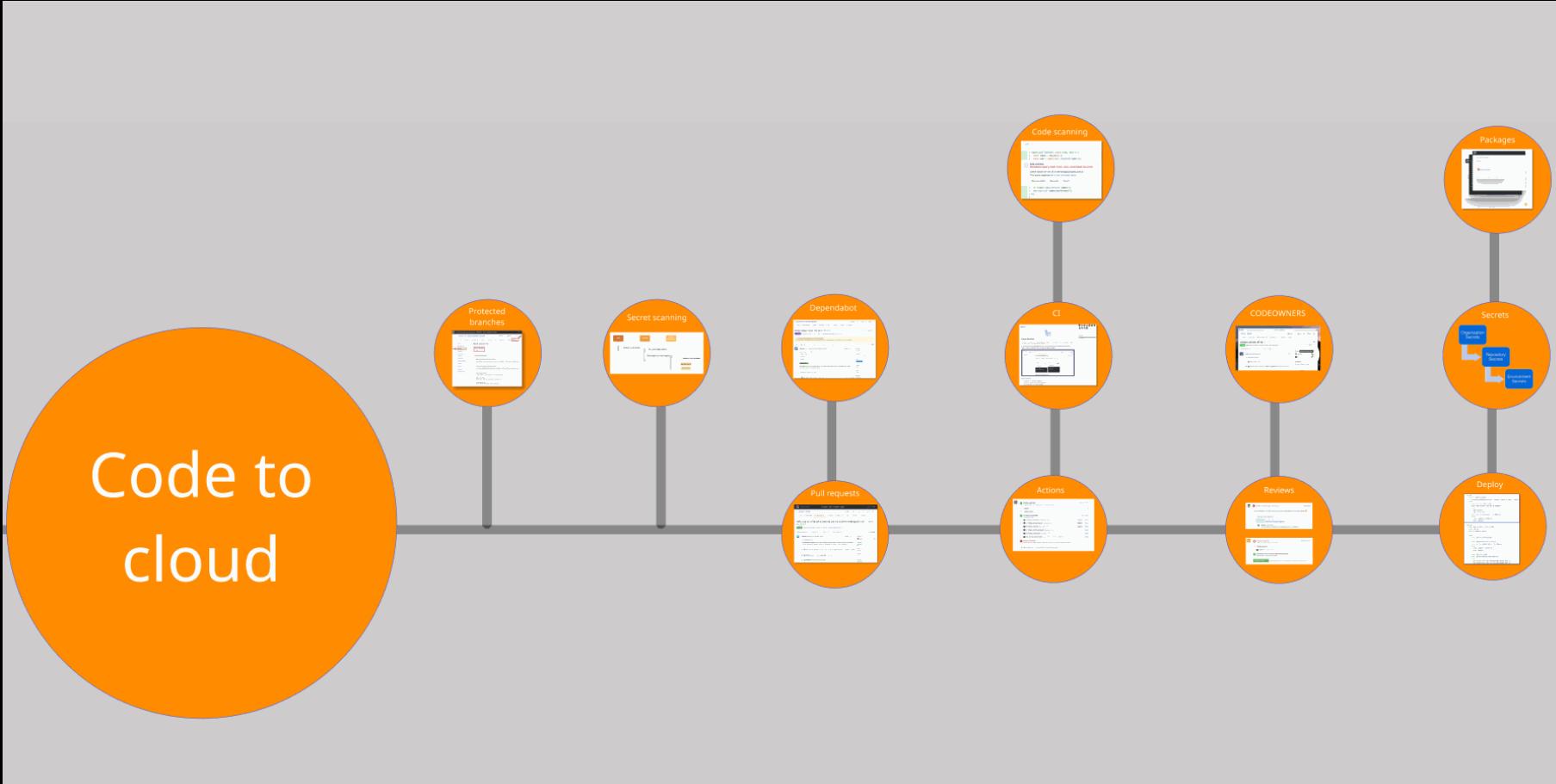
- Ideal for collaboration
- Always choose this whenever possible
- Protected branches ensure security

## Fork:

- Required to collaborate without “write” access to repo
- Often used in open source because of access management

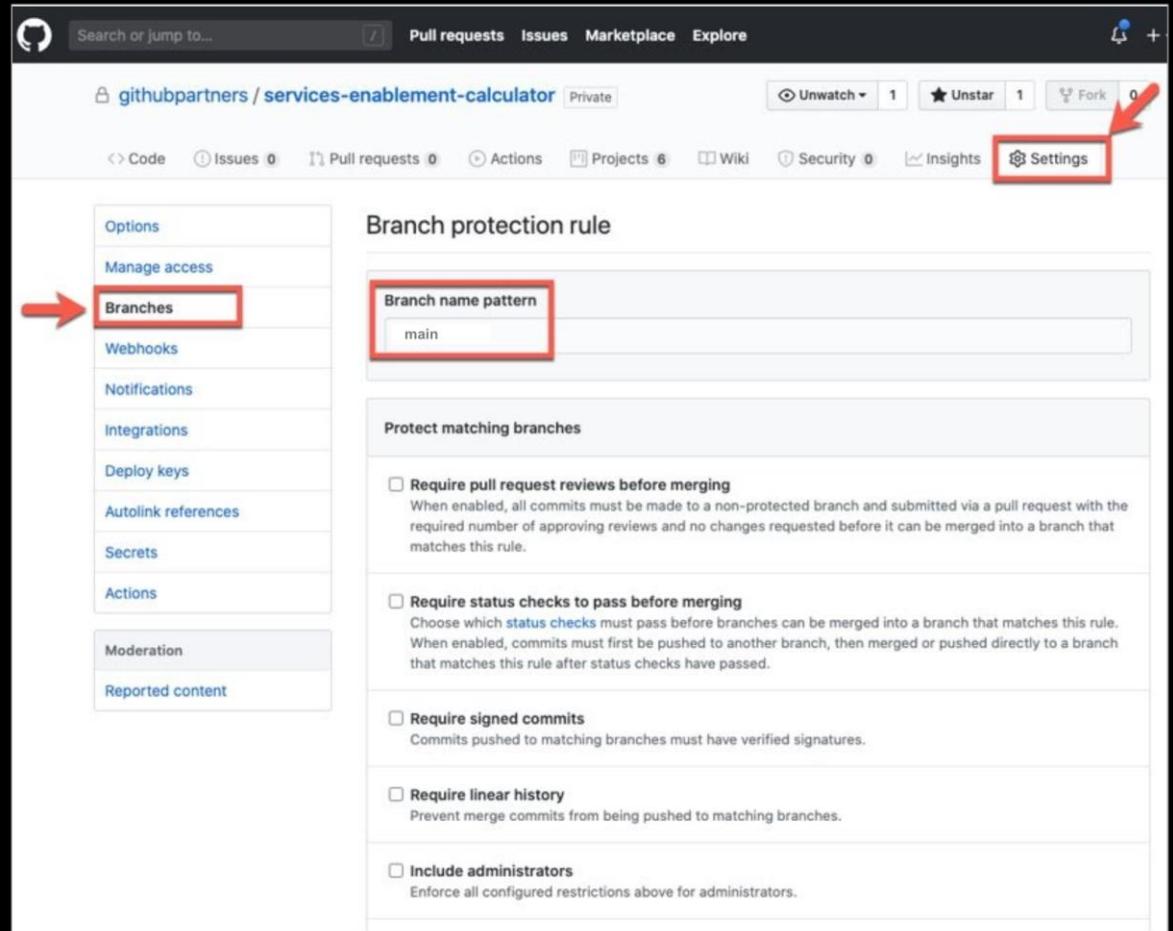


# About protected branches

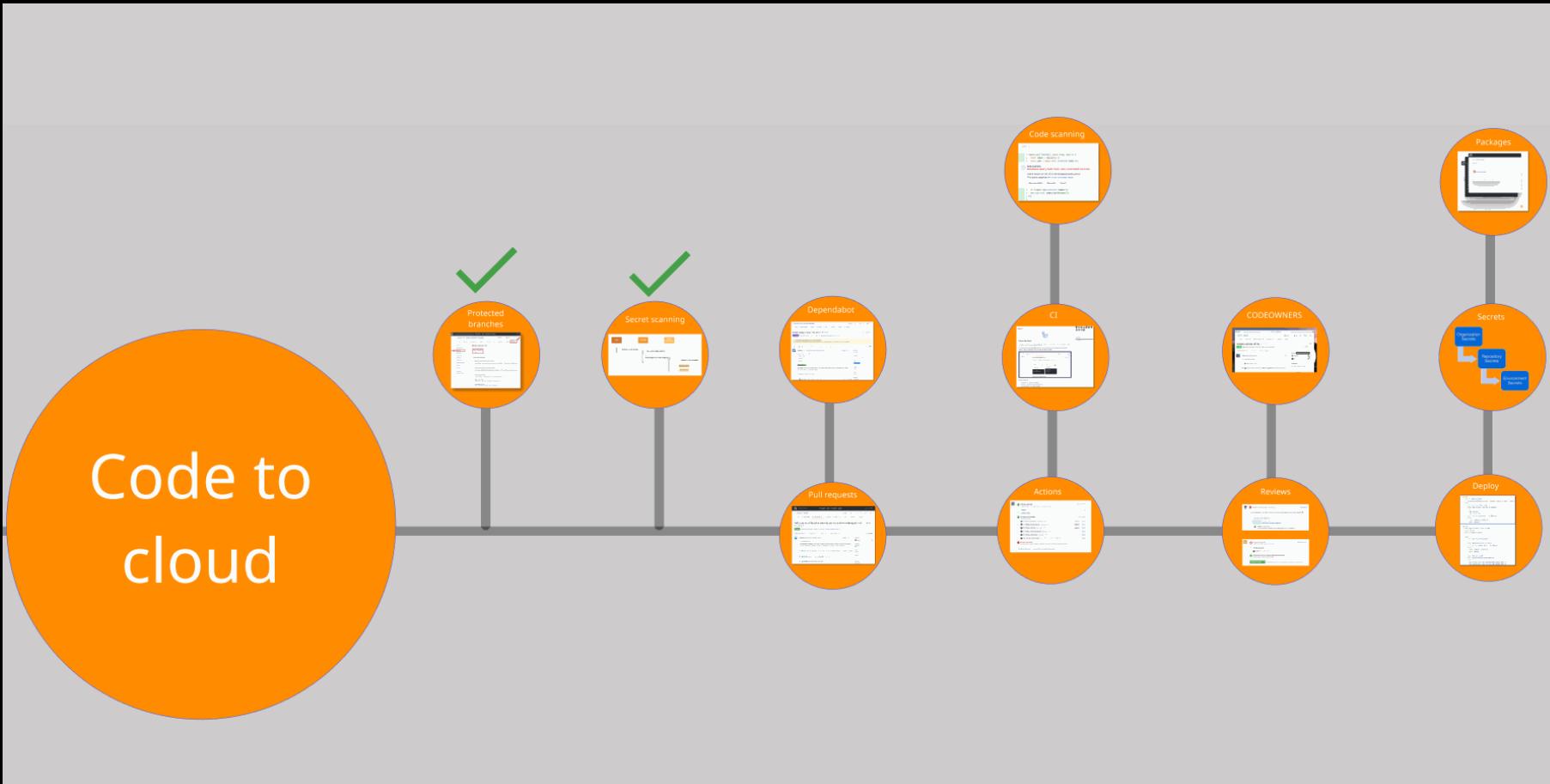


# Branch protection implementation

- Repository settings
- Admins only
- Name pattern options
- Limit merging & committing, not creating



# A Pull Request



# Pull Request timeline

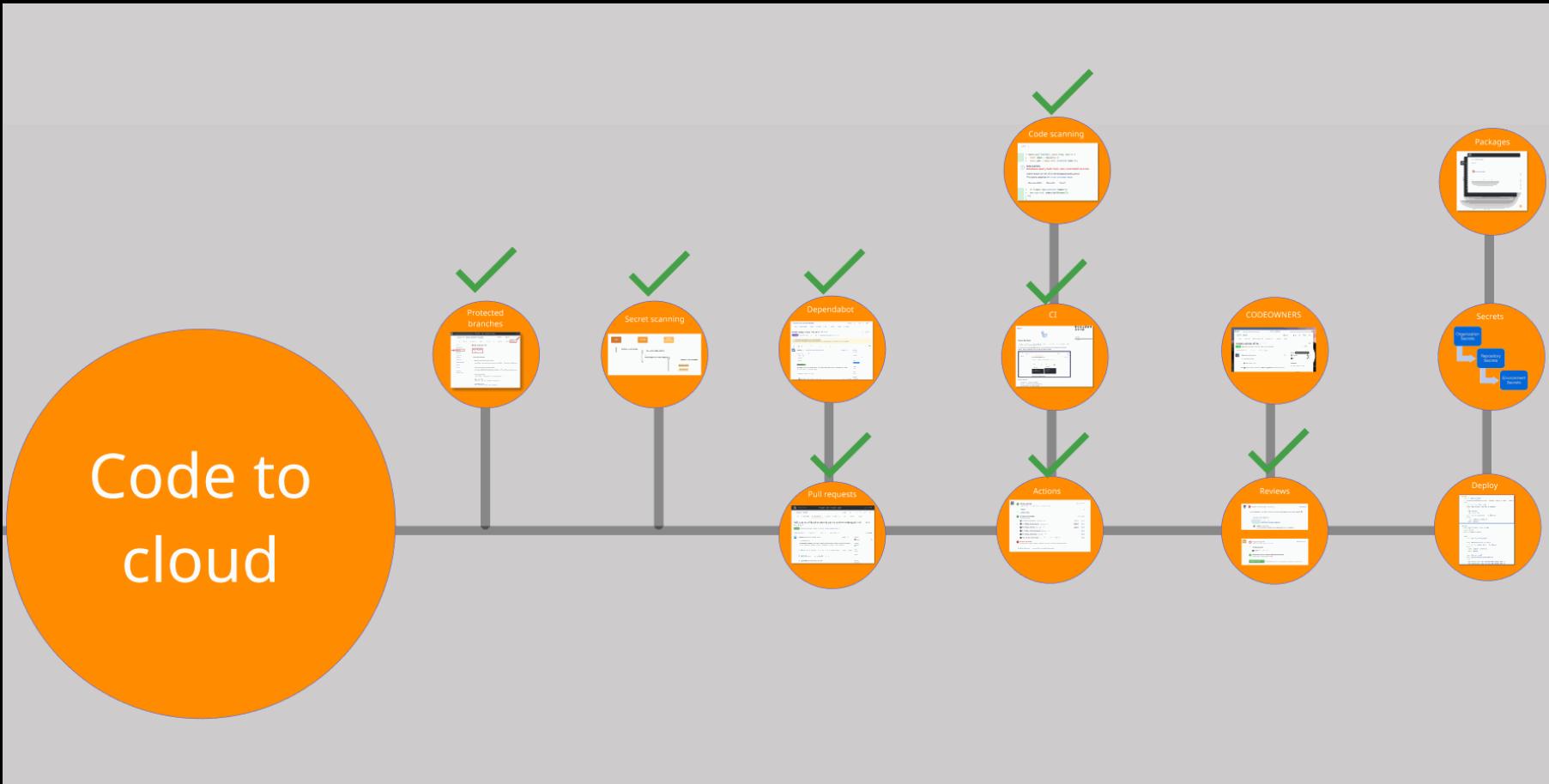
A screenshot of a GitHub pull request page for the `microsoft/vscode` repository. The pull request is titled "Add a cache to the editor override service to allow awaiting ext host" and is identified by the number #122067. The status is "Open".  
The main content area shows a conversation between `Iramos15` and `bpasero`. `Iramos15` has commented twice:

- The first comment fixes issue #116259 and describes implementing a suggestion from `@jrieken` to cache registered contributions before awaiting extension registration.
- The second comment requests a review from `bpasero`.

`Iramos15` has also self-assigned the pull request.  
On the right side, there are sections for Reviewers (`bpasero`), Assignees (`Iramos15`), Labels (None yet), Projects (None yet), and Milestone (No milestone).  
A red arrow points to the first comment from `Iramos15`, highlighting the fix for issue #116259.



# Codeowners



# Migrating a repository from Azure DevOps to GitHub



Hands-on lab 1:

[Migrating a repository from Azure DevOps to GitHub](#)

Time:

15 minutes

<https://github.com/XpiritCommunityEvents/HOL>

# Codespaces



# CodeSpaces

Desktop  Browser

The screenshot shows the Microsoft CodeSpaces interface. At the top, there are tabs for "Desktop" and "Browser", with "Desktop" being selected. On the left, there's a sidebar with a "CPU up to 32 cores" badge and a "DEBUG CONSOLE" section showing a list of manual sync operations. The main area is a dark-themed code editor with an "EXPLORER" sidebar showing a GitHub repository structure. The code editor displays a file named "emotion.rb". Below the code editor is a "TERMINAL" tab showing logs of extension installations. At the bottom, there are sections for "PORTS" (listing web, hmr, mysql, and api ports) and "TERMINAL", "PROBLEMS", "OUTPUTS", and "DEBUG CONSOLE" tabs. A GitHub logo is in the bottom right corner.

CPUs up to  
**32 cores**

DEBUG CONSOLE

TestSynchronizer - Manual sync

- ✓ preview
- ✓ preview -> merge
- ✓ preview -> accept
- ✓ preview -> merge -> accept

Codespaces

16.28% 2.11 GHz 100% 8.95/15.93 GB C: 74.18% remaining

TERMINAL

```
[09:43:36] Starting 'watch-extension:vscode-api-tests'...
[09:43:36] Finished 'clean-extension:typescript-language-features' after 248 ms
[09:43:36] Starting 'watch-extension:typescript-language-features'...
[09:43:36] Finished 'clean-extension:php-language-features' after 384 ms
[09:43:36] Starting 'watch-extension:php-language-features'...
[09:43:40] Finished 'clean-extension:html-language-features-server' after 4.66 s
[09:43:40] Starting 'watch-extension:html-language-features-server'...
[09:43:43] Starting 'clean-client' after 7.33 s
[09:43:43] Starting 'watch-client'...
[09:44:50] [monaco.d.ts] Starting monaco.d.ts generation
[09:44:56] [monaco.d.ts] Finished monaco.d.ts generation
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: zsh

PORTS 4

Port

- web (3000)
- hmr (55306)
- mysql (3306)
- api (3001)

GitHub

# Pricing



Available today for GitHub Teams and GitHub Enterprise Cloud customers

[Get started >](#)

Try Codespaces today in free trial until September 10, 2021

## Codespaces Compute

Isolated VMs billed per second

Cores	RAM	Price
2 core	4GB	\$18 per hour
4 core	8GB	\$36 per hour
8 core	16GB	\$72 per hour
16 core	32GB	\$1.44 per hour
32 core	64GB	\$2.88 per hour

## Codespaces storage

charged when inactive

**\$0.07**

per gigabyte per month



# Setting up Codespaces to develop a web-app



Hands-on lab 2:

[Setting up Code Spaces to develop a web-app](#)

Time:

10 minutes

<https://github.com/XpiritCommunityEvents/HOL>

Put in  
correct spot  
in PPT



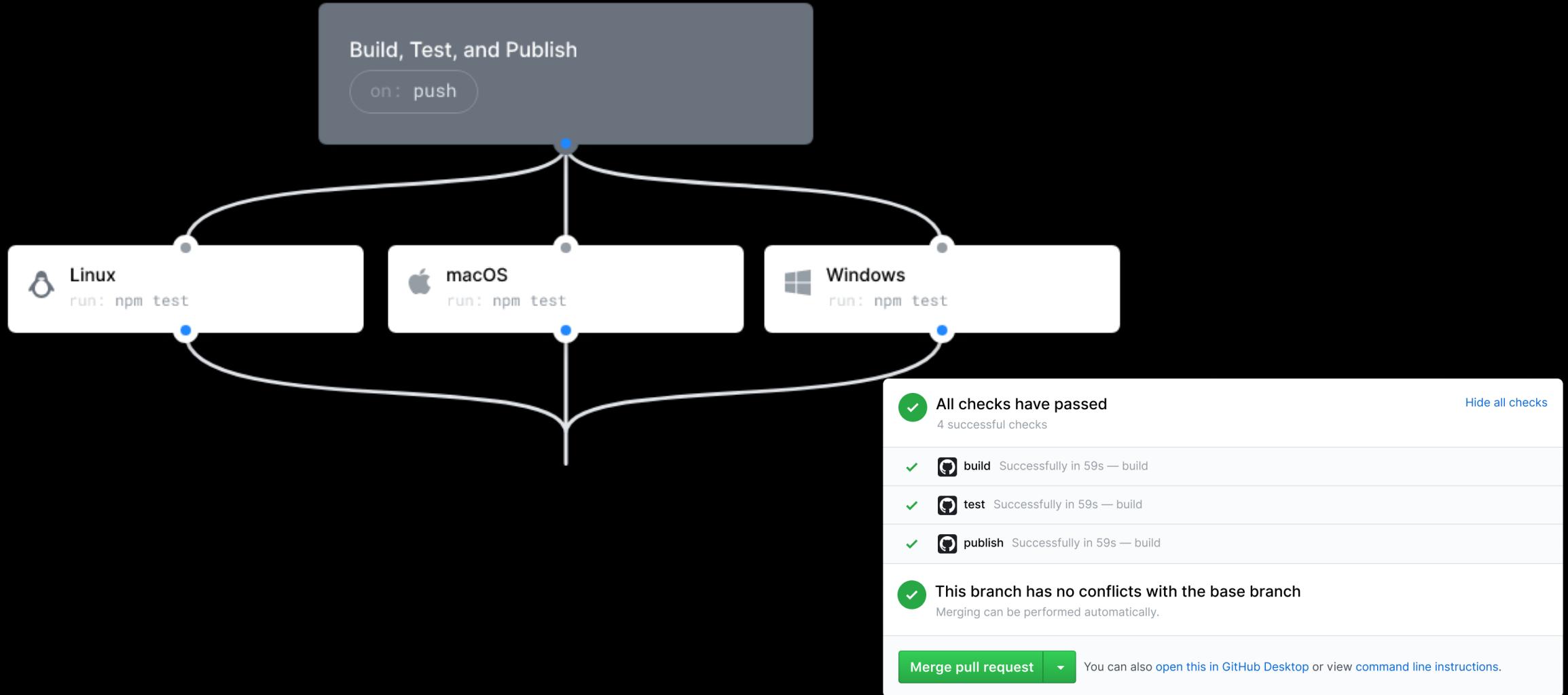
# ACTIONS



# What is GitHub Actions?

*“ GitHub Actions makes it easy to automate all your software workflows, now with world-class CI/CD. Build, test, and deploy your code right from GitHub. Make code reviews, branch management, and issue triaging work the way you want.*

Automate your workflow from idea to production





## Live Logs



**Linux, macOS,  
Windows, ARM,  
and containers**

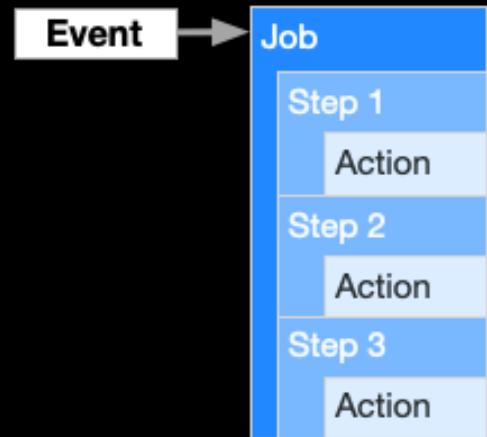


## Secret Store

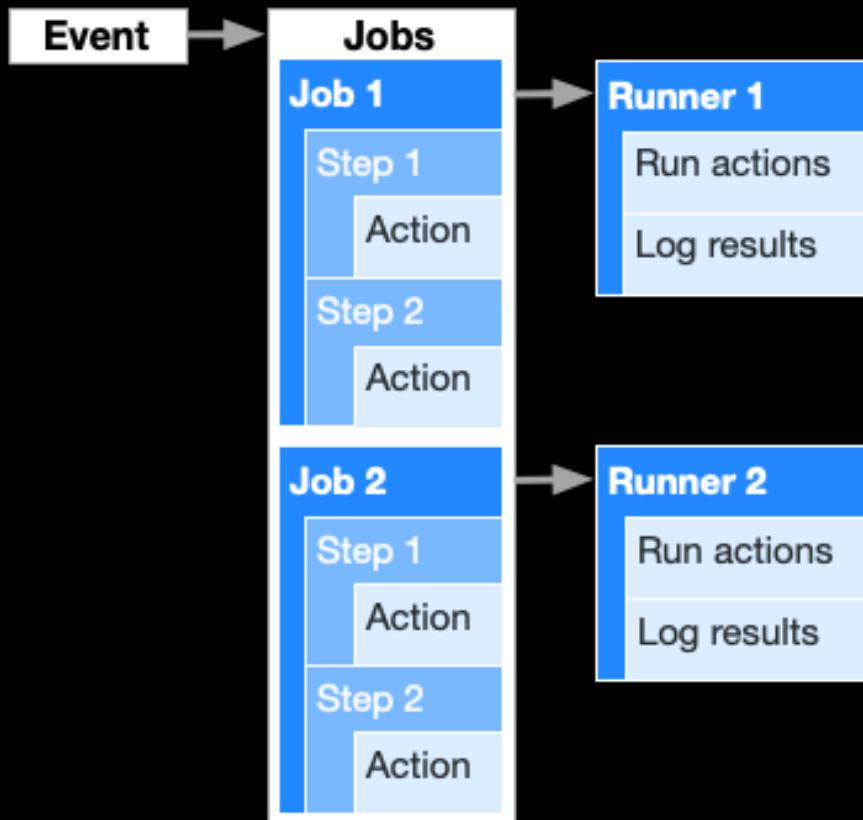


## Matrix Builds

# Overview



## Components of a workflow



Events:

Repo

Issues

Project

Pages

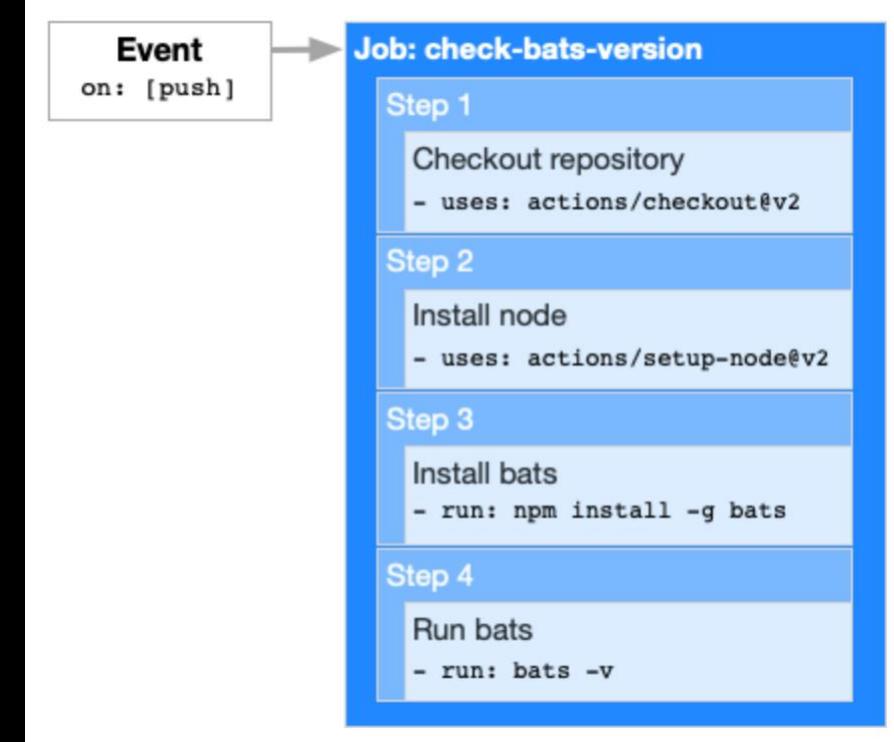
Wiki

Note:

When you use the GITHUB\_TOKEN to interact with GitHub, it will not trigger new events

# Yaml workflow definitions

```
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-node@v2
        with:
          node-version: '14'
      - run: npm install -g bats
      - run: bats -v
```



# Context & Expressions

A context provides access to a set of values that we can use in actions or in expressions

An Expression can be used to create more elaborate values or evaluate conditions

Each Step can have conditions that when evaluated to true will execute



# Context

Syntax:`${{ <context> }}`

`${{ context.ConstantName }}`

GitHub context provides access to: workflow, token, job, event, etc.

Environment provides access to environment variables

Secret provides access to the environment, organizational or repo secrets

Examples:

`${{ secrets.MYAPI_TOKEN }}`

`${{ secrets.AZUREAPPSERVICE_PUBLISHPROFILE }}`

"



# Expression

Syntax: \${{ <expression> }}

Expression:

steps:

- uses: actions/hello-world-javascript-action@v1.1  
if: \${{ <expression> }}

name: CI

on: push

jobs:

prod-check:

if: \${{ github.ref == 'refs/heads/main' }}

runs-on: ubuntu-latest

steps:

- run: echo "Deploying to production server on branch \$GITHUB\_REF"



# matrix

Create a job per item in the matrix  
e.g. for each language a CodeQL run

```
strategy:
  matrix:
    node: [10, 12, 14]
steps:
  # Configures the node version used on
  GitHub-hosted runners
  - uses: actions/setup-node@v2
    with:
      # The Node.js version to configure
      node-version: ${{ matrix.node }}
```



# Where do actions come from?

GitHub  
Community  
Marketplace

```
- name: Setup Node  
  uses: actions/setup-node@v1
```



<https://github.com/actions/setup-node>

```
- name: Run Azure webapp deploy  
  uses: azure/webapps-deploy@v2
```



<https://github.com/azure/webapps-deploy>



# Runners

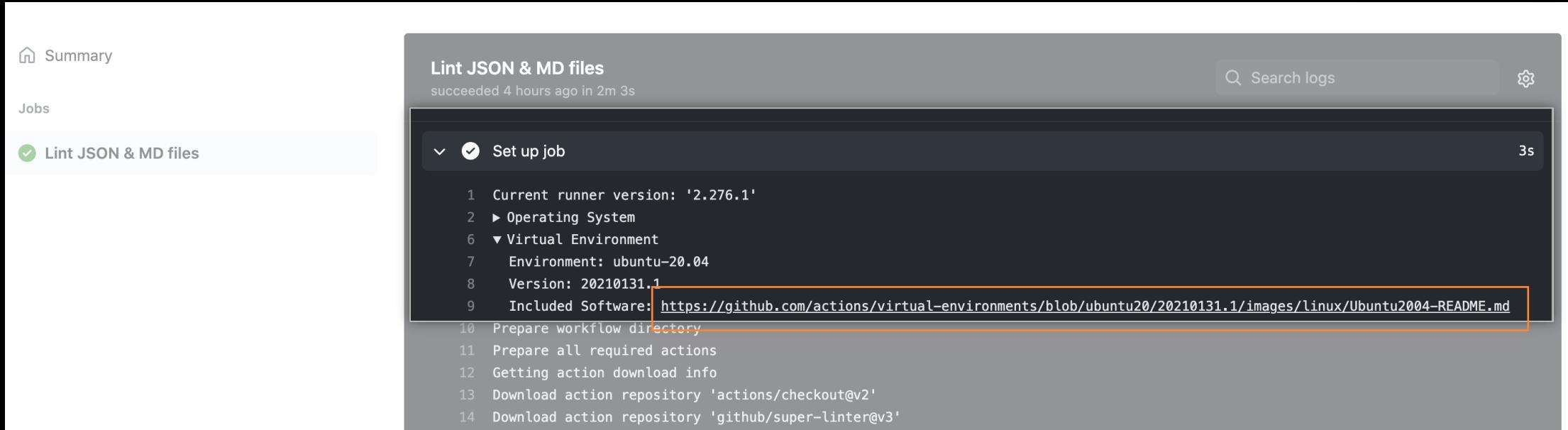
## GitHub Hosted

- Hosted on Azure (Standard\_DS2\_v2 virtual machines)
- Mac is hosted on GitHub's own macOS Cloud.
- The Linux and macOS virtual machines both run using passwordless sudo

**Note:** If you use an IP address allow list for your GitHub organization or enterprise account, you cannot use GitHub-hosted runners and must instead use self-hosted runners. For more information, see "[About self-hosted runners](#)."

# Pre-installed software

Look at the install logs of Set up job step:



The screenshot shows the GitHub Actions interface. On the left, there's a sidebar with 'Summary', 'Jobs', and a selected 'Lint JSON & MD files' job. The main area displays the 'Lint JSON & MD files' job summary: 'succeeded 4 hours ago in 2m 3s'. Below this is the log output for the 'Set up job' step. The log entries are numbered 1 through 14. Entry 9 highlights the inclusion of pre-installed software with a URL: <https://github.com/actions/virtual-environments/blob/ubuntu20/20210131.1/images/linux/Ubuntu2004-README.md>. This URL is highlighted with an orange rectangle.

```
1 Current runner version: '2.276.1'  
2 ▶ Operating System  
6 ▼ Virtual Environment  
7 Environment: ubuntu-20.04  
8 Version: 20210131.1  
9 Included Software: https://github.com/actions/virtual-environments/blob/ubuntu20/20210131.1/images/linux/Ubuntu2004-README.md  
10 Prepare workflow directory  
11 Prepare all required actions  
12 Getting action download info  
13 Download action repository 'actions/checkout@v2'  
14 Download action repository 'github/super-linter@v3'
```

GitHub-hosted runners are updated weekly

If there is a tool that you'd like to request, please open an issue at  
[actions/virtual-environments](#)



# Available Runners (12-09-2021)

Virtual environment	YAML workflow label	Notes
Windows Server 2022 <sup>[beta]</sup>	windows-2022	The windows-latest label currently uses the Windows Server 2019 runner image.
Windows Server 2019	windows-latest or windows-2019	
Windows Server 2016	windows-2016	
Ubuntu 20.04	ubuntu-latest or ubuntu-20.04	
Ubuntu 18.04	ubuntu-18.04	
Ubuntu 16.04 <sup>[deprecated]</sup>	ubuntu-16.04	Deprecated and limited to existing customers only. Migrate to Ubuntu 20.04. For more information, see <a href="#">the blog post</a> .
macOS Big Sur 11	macos-11	The macos-latest label currently uses the macOS 10.15 runner image.
macOS Catalina 10.15	macos-latest or macos-10.15	



# Self Hosted Runners

You can add self-hosted runners at various levels in the management hierarchy:

- **Repository-level** runners are dedicated to a single repository.
- **Organization-level** runners can process jobs for multiple repositories in an organization.
- **Enterprise-level** runners can be assigned to multiple organizations in an enterprise account



# Communication between self-hosted runners and GitHub

Uses HTTPS Long-poll with 50s timeout

If you use an IP address allow list for your GitHub organization or enterprise account, you must add your self-hosted runner's IP address to the allow list

Note: Some of the domains listed below are configured using `CNAME` records. Some firewalls might require you to add rules recursively for all `CNAME` records. Note that the `CNAME` records might change in the future, and that only the domains listed below will remain constant.

```
github.com
api.github.com
*.actions.githubusercontent.com
github-releases.githubusercontent.com
github-registry-files.githubusercontent.com
codeload.github.com
*.pkg.github.com
pkg-cache.githubusercontent.com
pkg-containers.githubusercontent.com
pkg-containers-az.githubusercontent.com
*.blob.core.windows.net
```



# Self hosted and public repos

People love ways to run bitcoin miners!

Recommended to only use self hosted runners on private repos

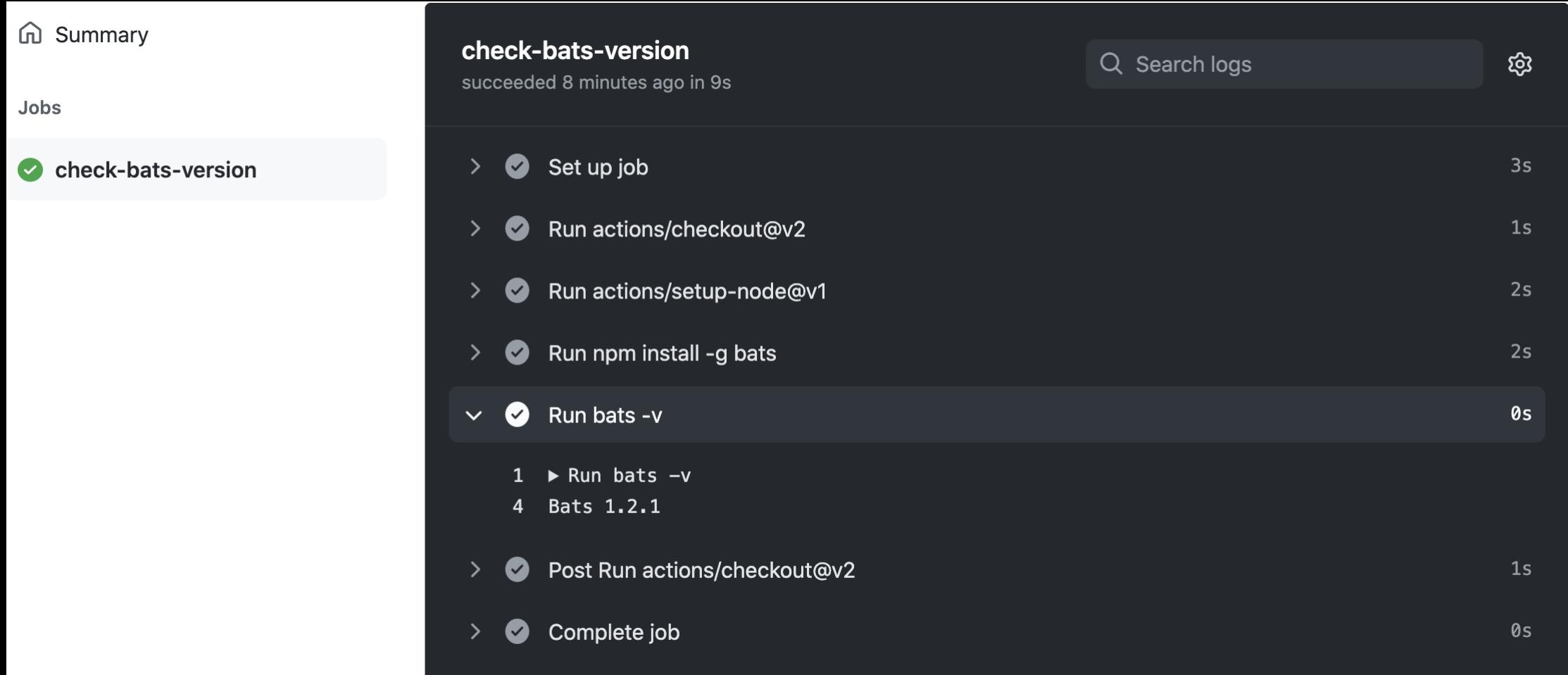
Forks of your repo, can otherwise run dangerous workflows (via pull request)!

Untrusted workflows running on your self-hosted runner pose significant security risks for your machine and network environment, especially if your machine persists its environment between jobs. Some of the risks include:

- Malicious programs running on the machine.
- Escaping the machine's runner sandbox.
- Exposing access to the machine's network environment.
- Persisting unwanted or dangerous data on the machine.



# Job runner logs



The screenshot shows the GitHub Actions interface for a job named "check-bats-version". The job status is "succeeded 8 minutes ago in 9s". The logs are displayed in a tree-like structure:

- > ✓ Set up job 3s
- > ✓ Run actions/checkout@v2 1s
- > ✓ Run actions/setup-node@v1 2s
- > ✓ Run npm install -g bats 2s
- > ✓ Run bats -v 0s
  - 1 ► Run bats -v
  - 4 Bats 1.2.1
- > ✓ Post Run actions/checkout@v2 1s
- > ✓ Complete job 0s

A search bar at the top right says "Search logs" and a settings gear icon is also present.



# Compare AzDo <>> GitHub Yaml

## Azure Pipelines

```
jobs:  
  - job: scripts  
    pool:  
      vmImage: 'windows-latest'  
    steps:  
      - script: echo "This step runs in the default shell"  
      - bash: echo "This step runs in bash"  
      - pwsh: Write-Host "This step runs in PowerShell Core"  
      - task: PowerShell@2  
        inputs:  
          script: Write-Host "This step runs in PowerShell"
```

## GitHub Actions

```
jobs:  
  scripts:  
    runs-on: windows-latest  
    steps:  
      - run: echo "This step runs in the default shell"  
      - run: echo "This step runs in bash"  
        shell: bash  
      - run: Write-Host "This step runs in PowerShell Core"  
        shell: pwsh  
      - run: Write-Host "This step runs in PowerShell"  
        shell: powershell
```

More information: <https://docs.github.com/en/actions/migrating-to-github-actions/migrating-from-azure-pipelines-to-github-actions>



# Pricing

## Private repositories

### Included minutes

Free	2,000
	minutes per month

Pro	3,000
	minutes per month

Team	3,000
	minutes per month

Enterprise	50,000
	minutes per month

### Additional hosted runner minutes

Linux	\$0.008
2 cores, 7GB	per minute

Windows	\$0.016
2 cores, 7GB	per minute

macOS	\$0.08
2 cores, 7GB	per minute

Self-hosted	Free
-------------	------

Included, hosted runner minutes are consumed at different rates for each operating system. GitHub Actions is not available for private repos in legacy per-repository plans. [Learn more](#)

# Creating your first Actions workflow



Hands-on lab 3:

[Creating your first Action Workflow](#)

Time:

15 minutes

<https://github.com/XpiritCommunityEvents/HOL>

Put in  
correct spot  
in PPT

# Creating a .NET Actions workflow



Hands-on lab 4:

[Creating a .NET Actions workflow](#)

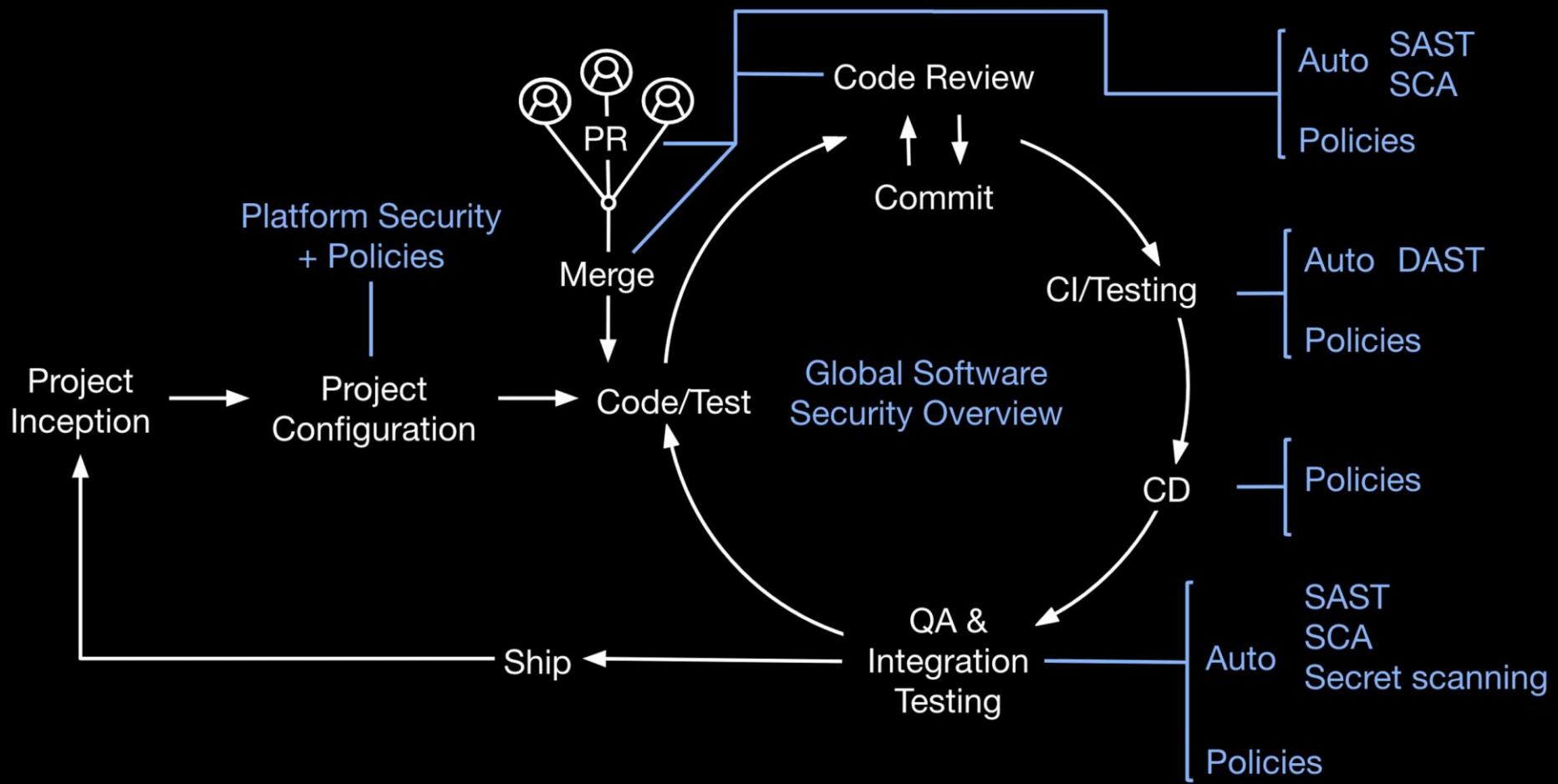
Time:

20 minutes

Put in  
correct spot  
in PPT

# Security

Security is not separate. It's deeply ingrained throughout our development process.



# GitHub's Advanced Security capabilities



## Supply Chain

- **Dependency review**

Identify new dependencies and vulnerabilities in a PR



## Code

- **Code scanning**

Static analysis of every git push powered by CodeQL and extensible with other scanning engines

- **Secret scanning**

Automatic notifications of API tokens exposed in git history



## Development Lifecycle

- **Security overview**

View security results of all kinds across your organization



# Code scanning

CodeQL analysis consists of three steps:

- 1.Preparing the code, by creating a CodeQL database
- 2.Running CodeQL queries against the database
- 3.Interpreting the query results



# Supported languages

Language	Variants	Compilers	Extensions
<b>C/C++</b>	C89, C99, C11, C18, C++98, C++03, C++11, C++14, C++17, C++20 <a href="#">[1]</a>	Clang (and clang-cl <a href="#">[2]</a> ) extensions (up to Clang 12.0), GNU extensions (up to GCC 11.1), Microsoft extensions (up to VS 2019), Arm Compiler 5 <a href="#">[3]</a>	.cpp, .c++, .cxx, .hpp, .hh, .h++, .hxx, .c, .cc, .h
<b>C#</b>	C# up to 9.0	Microsoft Visual Studio up to 2019 with .NET up to 4.8, .NET Core up to 3.1 .NET 5	.sln, .csproj, .cs, .cshtml, .xaml
<b>Go (aka Golang)</b>	Go up to 1.16	Go 1.11 or more recent	.go
<b>Java</b>	Java 7 to 16 <a href="#">[4]</a>	javac (OpenJDK and Oracle JDK), Eclipse compiler for Java (ECJ) <a href="#">[5]</a>	.java
<b>JavaScript</b>	ECMAScript 2021 or lower	Not applicable	.js, .jsx, .mjs, .es, .es6, .htm, .html, .xhm, .xhtml, .vue, .json, .yaml, .yml, .raml, .xml <a href="#">[6]</a>
<b>Python</b>	2.7, 3.5, 3.6, 3.7, 3.8, 3.9	Not applicable	.py
<b>TypeScript</b> <a href="#">[7]</a>	2.6-4.2	Standard TypeScript compiler	.ts, .tsx



# Enable Code Scanning

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Autolink references

Actions

Environments

Secrets

Pages

## Configure security and analysis features

Security and analysis features help keep your repository secure and updated. By enabling these features, you're granting us permission to perform read-only analysis on your repository.

**Dependency graph**  
Understand your dependencies. Enable

**Dependabot alerts**  
Receive alerts of new vulnerabilities that affect your dependencies. Enable

**Dependabot security updates**  
Easily upgrade to non-vulnerable dependencies. Enable

## GitHub Advanced Security

GitHub Advanced Security features are billed per active committer in private repositories. [Learn more](#).

**Code scanning**  
Automatically detect common vulnerabilities and coding errors. Set up

**Secret scanning**  
Receive alerts when secrets or keys are checked in. Enable



# Enable Code Scanning

- Overview
- Security policy
- Security advisories
- Dependabot alerts
- Code scanning alerts
- Secret scanning alerts

Get started with code scanning

Automatically detect common vulnerabilities and coding errors

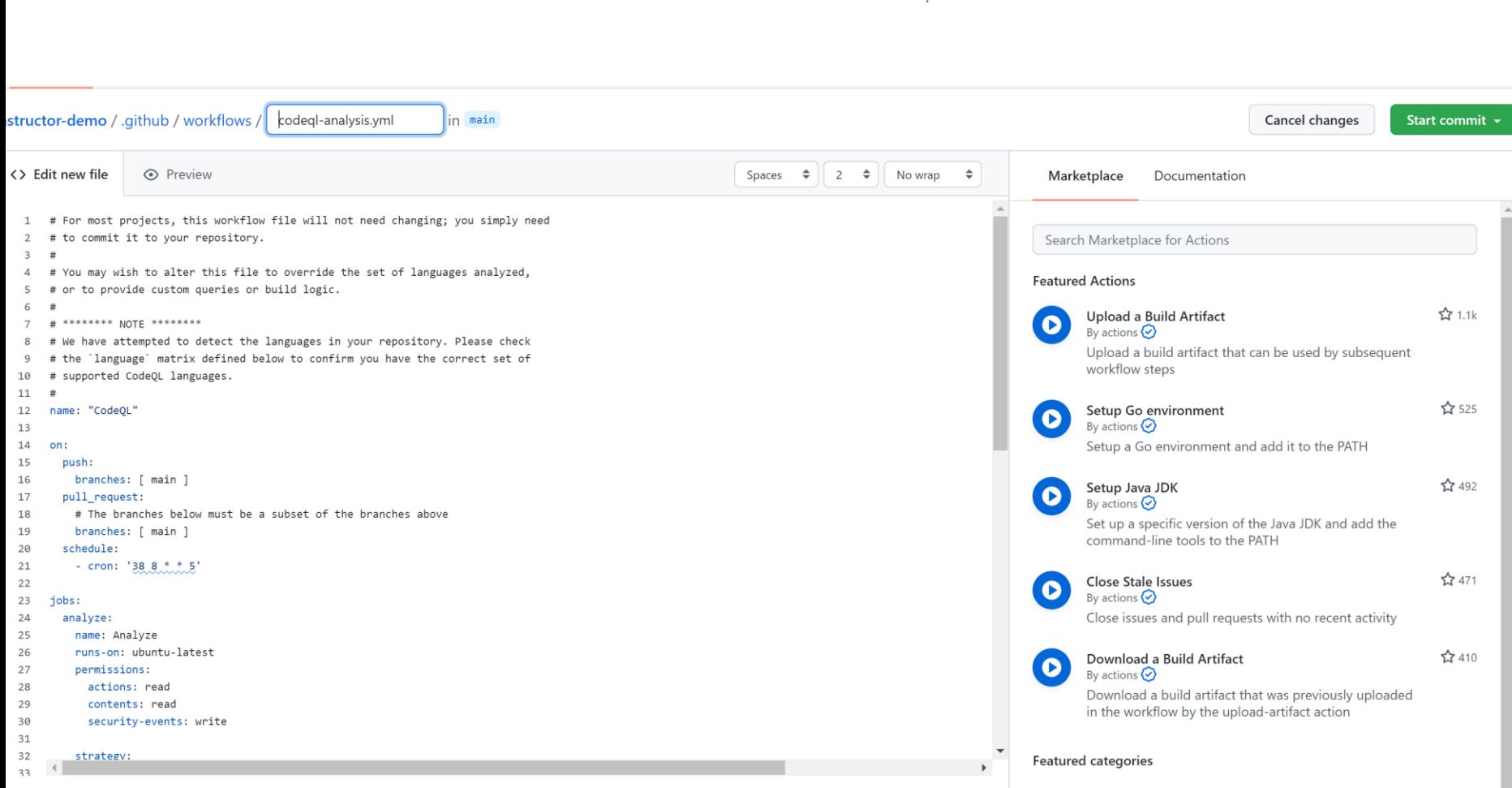
**CodeQL Analysis**  
by GitHub 

Security analysis from GitHub for C, C++, C#, Java, JavaScript, TypeScript, Python, and Go developers.

[Set up this workflow](#)



# Enable Code Scanning



The screenshot shows a GitHub Actions workflow configuration page. The workflow file is named `codeql-analysis.yml` and is located in the `main` branch of the repository `structor-demo / .github / workflows`. The code editor displays the YAML content of the workflow file, which defines a job named "Analyze" that runs on an Ubuntu latest container, with specific permissions for actions, contents, and security events.

```
1 # For most projects, this workflow file will not need changing; you simply need
2 # to commit it to your repository.
3 #
4 # You may wish to alter this file to override the set of languages analyzed,
5 # or to provide custom queries or build logic.
6 #
7 # ***** NOTE *****
8 # We have attempted to detect the languages in your repository. Please check
9 # the 'language' matrix defined below to confirm you have the correct set of
10 # supported CodeQL languages.
11 #
12 name: "CodeQL"
13
14 on:
15   push:
16     branches: [ main ]
17   pull_request:
18     # The branches below must be a subset of the branches above
19     branches: [ main ]
20   schedule:
21     - cron: '38 8 * * 5'
22
23 jobs:
24   analyze:
25     name: Analyze
26     runs-on: ubuntu-latest
27     permissions:
28       actions: read
29       contents: read
30       security-events: write
31
32 strategy:
```

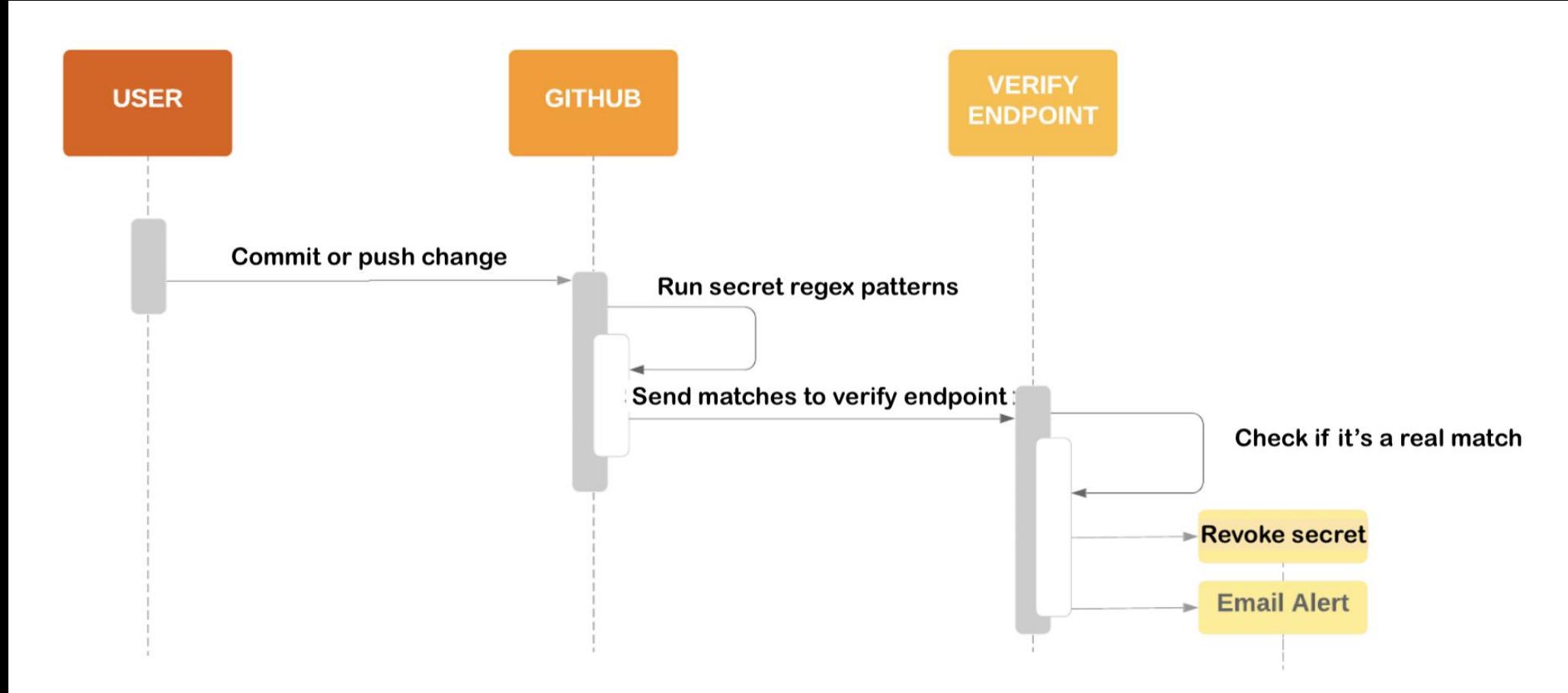
On the right side of the screen, there is a sidebar for the GitHub Marketplace. It features a search bar labeled "Search Marketplace for Actions" and a section titled "Featured Actions" containing five items:

- Upload a Build Artifact** (By actions) - 1.1k stars
- Setup Go environment** (By actions) - 525 stars
- Setup Java JDK** (By actions) - 492 stars
- Close Stale Issues** (By actions) - 471 stars
- Download a Build Artifact** (By actions) - 410 stars

Below the featured actions, there is a section titled "Featured categories".



# Secret scanning on every pushed commit



# Enable secret scanning

Configure security and analysis features

Security and analysis features help keep your repository secure and updated. By enabling these features, you're granting us permission to perform read-only analysis on your repository.

**Dependency graph**  
Understand your dependencies. Enable

**Dependabot alerts**  
Receive alerts of new vulnerabilities that affect your dependencies. Enable

**Dependabot security updates**  
Easily upgrade to non-vulnerable dependencies. Enable

**GitHub Advanced Security** Enable

GitHub Advanced Security features are billed per active committer in private repositories. [Learn more](#).

**Code scanning**  
Automatically detect common vulnerabilities and coding errors. Set up

**Secret scanning**  
Receive alerts when secrets or keys are checked in. Enable



# Setting up Code Scanning and Security Scanning for your repository



Hands-on lab 5:

[Setting up Code Scanning and Security Scanning for your repository](#)

Time:

20 minutes

<https://github.com/XpiritCommunityEvents/HOL>

Put in  
correct spot  
in PPT

# GitHub's Advanced Security capabilities



## Supply Chain

- **Dependency review**

Identify new dependencies and vulnerabilities in a PR



## Code

- **Code scanning**

Static analysis of every git push powered by CodeQL and extensible with other scanning engines

- **Secret scanning**

Automatic notifications of API tokens exposed in git history



## Development Lifecycle

- **Security overview**

View security results of all kinds across your organization



# Dependabot

Analyze Dependency Graph

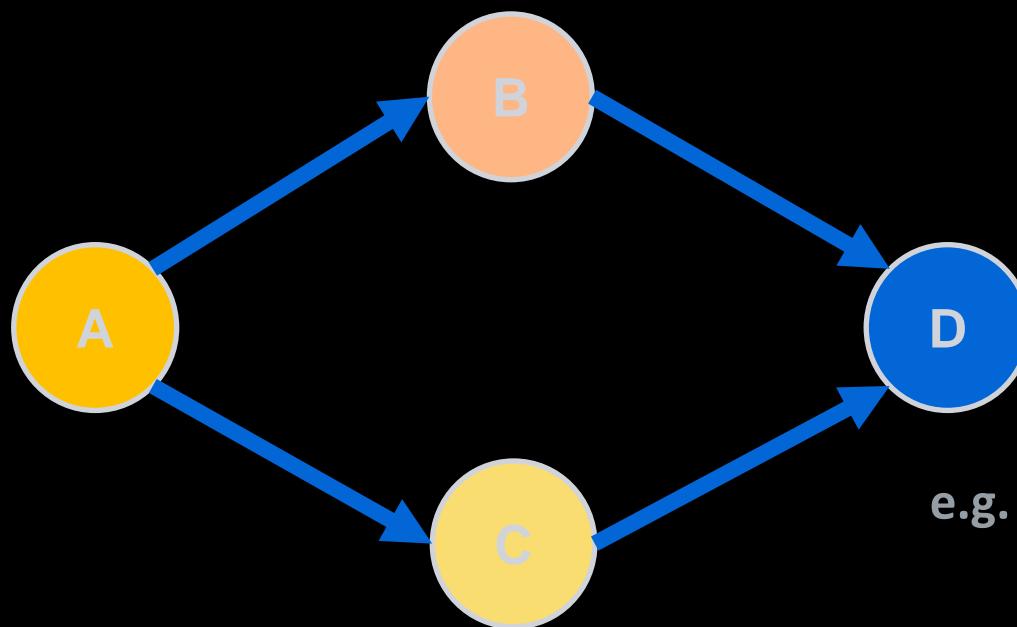
Generate pull request to update to new version

Generate pull request when known vulnerability



# Dependency Management

## Dependency Hell 101



e.g. `Newtonsoft.Json`



# Semantic Versioning

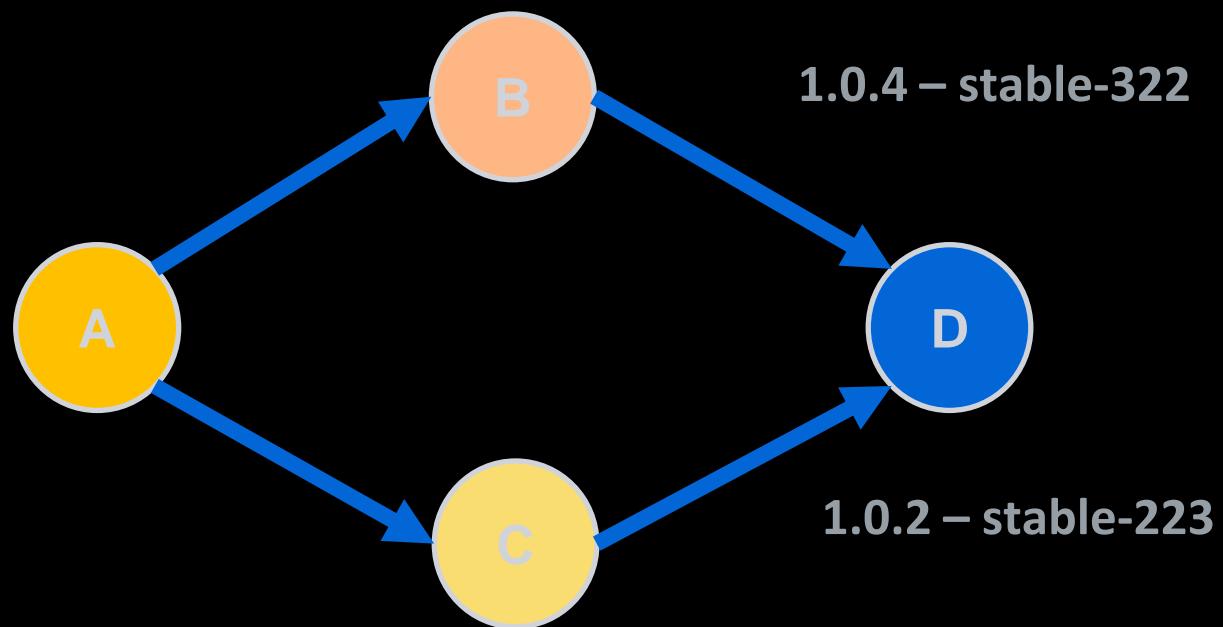


Given a version number **MAJOR.MINOR.PATCH**, increment the:

- **MAJOR** version when you make incompatible API changes,
- **MINOR** version when you add functionality in a backwards-compatible manner, and
- **PATCH** version when you make backwards-compatible bug fixes.
- Additional labels for pre-release and build metadata are available as extensions to the **MAJOR.MINOR.PATCH** format.

# Dependency Management

## Dependency Hell 101



# Enable Dependabot

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Configure security and analysis features

Security and analysis features help keep your repository secure and updated. By enabling these features, you're granting us permission to perform read-only analysis on your repository.

**Dependency graph**  
Understand your dependencies. Enable

**Dependabot alerts**  
Receive alerts of new vulnerabilities that affect your dependencies. Enable

**Dependabot security updates**  
Easily upgrade to non-vulnerable dependencies. Enable

# Enabling and using Dependabot on your repository (If time permits)



Hands-on lab 6:

[Enabling and using Dependabot on your repository](#)

Time:

15 minutes

Put in  
correct spot  
in PPT

# GitHub Code to Cloud Workshop Agenda

	<b>Introduction to GitHub</b>	13.30 – 14.00
	Migrating a repository from Azure DevOps to GitHub	14.00 – 14.15
	Codespaces – Your development IDE in the cloud	14.15 – 14.20
	Setting up Codespaces to develop a web-app	14.20 – 14.35
	GitHub Actions	14.35 – 15.00
	Coffee break	15.00 – 15.15
	Creating your first Actions workflow	15.15 – 15.30
	Creating a .NET Actions workflow	15.30 – 16.00
	GitHub Advanced Security	16.00 – 16.15
	Code Scanning and Secret Scanning	16.15 – 16.35
	Dependabot	16.35 – 16.50
	Wrap-up	16.50 – 17.00



# Marcel de Vries

Founder Xpirit  
Microsoft Regional Director, MVP  
[linkedin.com/in/marcelv](https://linkedin.com/in/marcelv)  
mdevries@xpirit.com



# Rob Bos

DevOps Consultant at Xpirit  
Microsoft MVP  
[linkedin.com/in/bosrob](https://linkedin.com/in/bosrob)  
rbos@xpirit.com



# Dennis Thie

Consultant at Xpirit  
[linkedin.com/in/dennisthie](https://linkedin.com/in/dennisthie)  
dthie@xpirit.com