

Ship Your Game With GitHub Actions



Let's connect!



@chtzvt



blog.ctis.me

Charlton Trezevant

Application Security Consultant

Xpirit USA

ctrezevant@xpirit.com



Agenda



All About GitHub Actions



CI & CD for Game Developers



Real-World Deep Dive

End Goals

Great Experiences for Developers and Players

1.

Reach players with a uniformly great game experience on as many platforms as possible.

2.

Consistent and fast releases of updates on all supported platforms simultaneously.

3.

Make the release process as simple and automatic as possible. Let developers focus on bugfixes and shipping new content.

All About Actions



Pipelines: GitHub Actions



Generic
workflow
engine:
automate
anything



Workflows =
text-based file
in your repo
(yaml)



Actions =
building blocks
in your
workflow



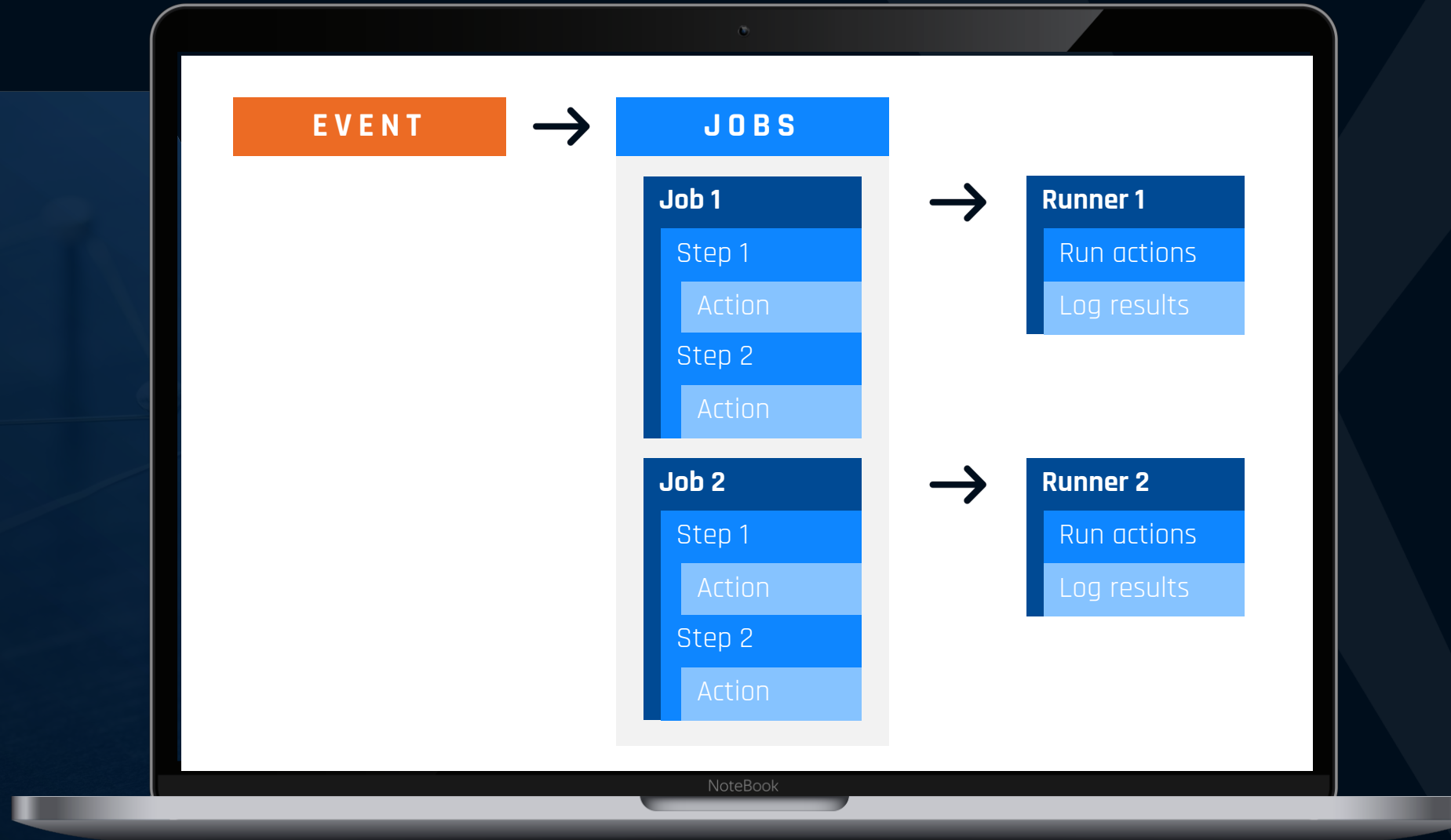
Jobs =
collection of
steps



Step = shell
script or calling
an action



Anatomy of a workflow



Pipelines: GitHub Actions



37+
triggers:

> Manual (workflow_dispatch)

> On: push

> On: pull_request

> On: issue

> On: page_created

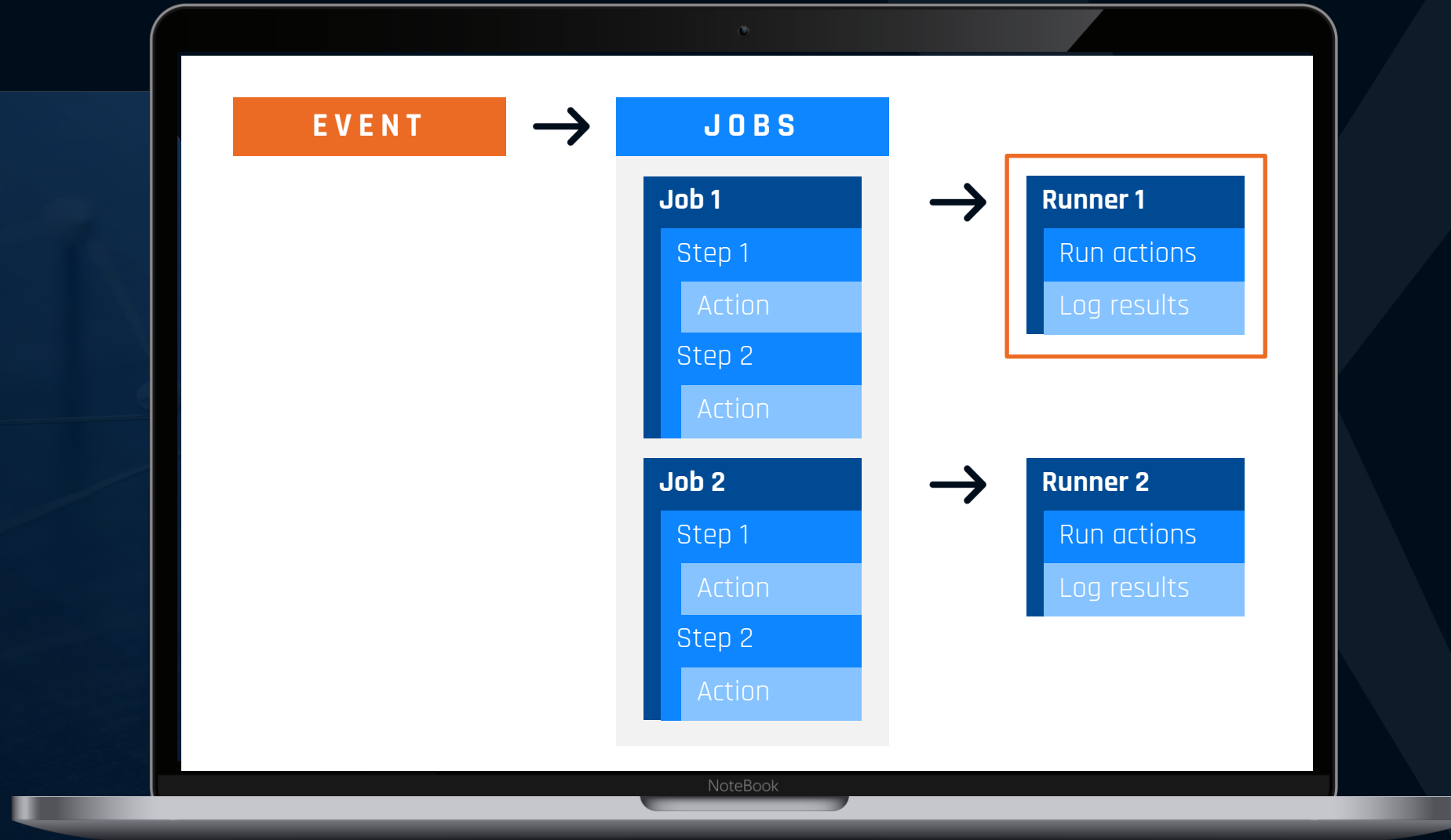
> All (repository_dispatch)

> On: issue_comment

> On: release

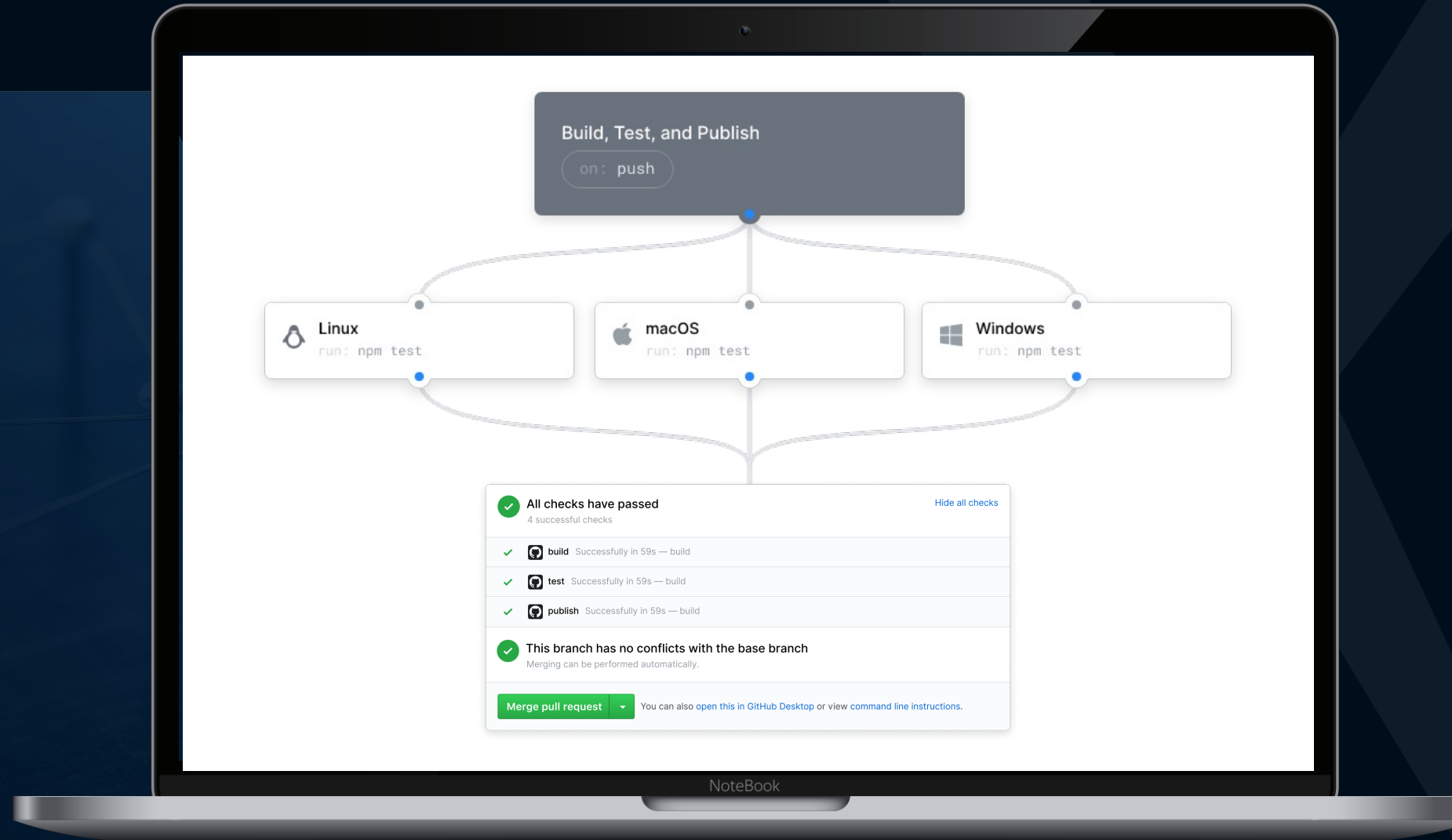


Anatomy of a workflow





Hosted environments



Functionality



Live
Logs



Secret
Store



Linux, macOS,
Windows,
ARM, and
containers



Matrix
Builds



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
```

EVENT
on: [push]



Job: check-bats-version

Step 1

Checkout repository
- uses: actions/checkout@v2

Step 2

Install node
- uses: actions/setup-node@v2

Step 3

Install bats
- run: npm install -g bats

Step 4

Run bats
- run: bats -v

NoteBook



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 }}`

NoteBook



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`"

NoteBook



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 }}
```

NoteBook

Where do actions come from?

- > **GitHub:** /github and /action orgs
- > **Community:** any public repo
- > **Marketplace:** 16k actions!

> name: Setup Node
uses: [actions/setup-node@v1](#)



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

> name: Run Azure webapp deploy
uses: [azure/webapps-deploy@main](#)



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

Runners

- > **GitHub Hosted**

 - > **Hosted on Azure (Standard_DS2_v2 virtual machines)**

 - > **GitHub managed, running in your VNET is on its way**
- > **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.](#)"

Runners - specs

Standard:

> Hardware specification for Windows and Linux virtual machines:

- > 2-core CPU (x86_64)
- > 7 GB of RAM
- > 14 GB of SSD space

> Hardware specification for macOS virtual machines:

- > 3-core CPU (x86_64)
- > 14 GB of RAM
- > 14 GB of SSD space

Larger runners (Beta)

Size (vcpu)	Memory (GB)	Storage (SSD)
4 cores	16 RAM	150 GB
8 cores	32 RAM	300 GB
16 cores	64 RAM	600 GB
32 cores	128 RAM	1200 GB
64 cores	256 RAM	2040 GB

Feature: GPU enabled runners

<https://github.com/github/roadmap/issues/505>





Available Runners (as per 12-03-2022)



Virtual environment	YAML workflow label	Notes
Windows Server 2022	<code>windows-latest</code> or <code>windows-2022</code>	The <code>windows-latest</code> label currently uses the Windows Server 2022 runner image.
Windows Server 2019	<code>windows-2019</code>	
Windows Server 2016 ^[deprecated]	<code>windows-2016</code>	Migrate to Windows 2019 or Windows 2022. For more information, see the blog post .
Ubuntu 20.04	<code>ubuntu-latest</code> or <code>ubuntu-20.04</code>	
Ubuntu 18.04	<code>ubuntu-18.04</code>	
macOS Big Sur 11	<code>macos-latest</code> or <code>macos-11</code>	The <code>macos-latest</code> label currently uses the macOS 11 runner image.
macOS Catalina 10.15	<code>macos-10.15</code>	



Pre-installed software

- > Look at the install logs of "Set up job" step

```
Lint JSON & MD files
succeeded 4 hours ago in 2m 3s

Jobs
Lint JSON & MD files

Set up job 3s
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](https://github.com/actions/virtual-environments)

NoteBook

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.



Self-hosted runners - Groups

- > **Grouping runners:**
 - > Can be connected during installation,
 - > Or moved afterwards to a different group
- > **Per group, you can give access to:**
 - > Repos
 - > Branches
 - > Workflows

Communication between self-hosted runners and GitHub

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
```



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



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'. On the left, a sidebar contains a 'Summary' section with a home icon and a 'Jobs' section listing the job with a green checkmark. The main area displays the job's status as 'succeeded 8 minutes ago in 9s' and a search bar for logs. Below this, a list of steps is shown, each with a chevron, a checkmark, the step name, and a duration. The step 'Run bats -v' is expanded, showing its output: '1 Run bats -v' and '4 Bats 1.2.1'.



Step	Duration
> ✓ 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

Demo: [github-actions-requests](#)



Pricing

Private repositories

INCLUDED MINUTES 		ADDITIONAL HOSTED RUNNER MINUTES 	
Free	2,000 minutes per month	Linux 2 cores, 7GB	\$0.008 per minute
Pro	3,000 minutes per month	Windows 2 cores, 7GB	\$0.016 per minute
Team	3,000 minutes per month	macOS 2 cores, 7GB	\$0.08 per minute
Enterprise	50,000 minutes per month	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](#)

Game Development



Game Development is Unique

> Engine-Centric Development Workflow

> Large Assets

> Complex Toolchain

> Long Build Times

> Native-Centric

- Large builds shipped to customers
- Diverse consumer hardware targets
- Many ways things can break/bug out
- Console builds are another beast entirely
- Storefronts are hard!



Engines are Special

- > Integrated IDEs where code is not a first-class citizen

- > Small changes often produce huge diffs

End Goals

Great Experiences for Developers and Players

1.

Reach players with a uniformly great game experience on as many platforms as possible.

2.

Consistent and fast releases of updates on all supported platforms simultaneously.

3.

Make the release process as simple and automatic as possible. Let developers focus on bugfixes and shipping new content.

Success

Oh for sure, it literally feels like I'm cheating only having to click like 2 buttons and having it do everything automatically. Can't thank you enough than I have already but I will continue thanking you regardless

If your job doesn't think these actions are fantastic Imma need a word with them



Our Challenges



Small Team, Limited Resources



Manage Complexity and Cost



Manage Releases on Tons of Platforms



Consistency & Reusability



Ensure Quality, Great Player Experience



Security

Scaling is Hard



$(\# \text{ builds}) * (\# \text{ stores}) +$
 $(\# \text{ console releases})$

Other Considerations

- "Anyone Can Deploy"
- Each store/target has a bespoke toolchain & release process
- Dependency management
- Isolate code issues from build issues
- Feature Flags & Embedded Build Information
 - Enabling/disabling Steam SDK
 - Performance optimizations
 - Testing features (noclip, etc)

Let's Dive In



Cost & Efficiency Tips



Avoid repeat work.
Share and use
caches wherever
possible



Use Linux runners.
Optimize build times,
concurrency, and
artifact retention



Avoid surprises.
Monitor and limit
spending in the
billing dashboard

Xpirit and GitHub



Our experts love GitHub:

<https://xpir.it/github>

GitHub - Community

- > **XPRT magazine articles**
- > **Ask the Expert sessions**
- > **Rob Bos: Conference talks at GitHub InFocus 2021, GitHub Universe 2021, NDC Security (2022), Techorama BE (2022), and others**
 - > How to use GitHub Actions with Security in mind
 - > Protect your code with GitHub security features (Code signing + Advanced Security)
- > **Rob Bos: LinkedIn Learning Course on GitHub Advanced Security**
- > **Book: Accelerate DevOps with GitHub (Michael)**

<https://xpir.it/github>

- > **Blog posts galore!**
 - > Best practices for Actions
 - > Internal Actions Marketplace
 - > Self-hosted runners on Kubernetes
 - > GitHub tokens and best practices
- > **GitHub workshop: Code to Cloud**
- > **GitHub Workshop: VisugXL BE**
- > **Pluralsight course**
- > **GitHub Community Stuttgart**

Xpirit is Hiring



Start Your Journey:

xpirit.com/dna/xpirit-usa

Thank You!

