

Authoring and Maintaining GitHub Actions

Introducing Custom Actions



Kamran Ayub

Helping you prepare for the GitHub Actions exam

www.linkedin.com/in/kamranayub



Authoring and Maintaining GitHub Actions

Version Check



Version Check



This course was created by using:

- Node.js v20 (LTS)
- Docker Compose v2 (Windows)
- Ubuntu 22.04 (Windows)



Version Check



This course is 100% applicable to:

- **Last 3 LTS versions of Node.js**
- **Last 3 LTS versions of Ubuntu**
- Docker Compose v2+



Not Applicable



This course is NOT applicable to:

- Any tooling that has reached its end-of-life date, if using GitHub-hosted runners



Installation References

Windows Subsystem for Linux

<https://learn.microsoft.com/en-us/windows/wsl/about>

Docker for Desktop

<https://www.docker.com/>

Node.js

<https://www.nodejs.org>

GitHub Actions Software Support Policy

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



Support Policies

Check Your End-Of-Life Date
<https://endoflife.date>

GitHub Actions Software Support Policy
<https://github.com/actions/runner-images>



Why Create Custom Actions?

What problems do they solve?



```
- name: Setup SSH passphrase
  env:
    SSH_PASSPHRASE: ${{secrets.VM_SSH_PASSPHRASE}}
    SSH_PRIVATE_KEY: ${{secrets.VM_SSH_KEY}}
  run: |
    ssh-agent -a $SSH_AUTH_SOCK >/dev/null
    echo 'echo $SSH_PASSPHRASE' >~/.ssh_askpass
    echo "$SSH_PRIVATE_KEY" | tr -d '\r' | DISPLAY=None SSH_ASKPASS=~/ssh_askpass ssh-add - >/dev/null
```

Example GitHub Action Step

What if you want to reuse this across workflows or projects?



```
- name: Setup SSH passphrase
  env:
    SSH_PASSPHRASE: ${{secrets.VM_SSH_PASSPHRASE}}
    SSH_PRIVATE_KEY: ${{secrets.VM_SSH_KEY}}
  run: |
    ssh-agent -a $SSH_AUTH_SOCK >/dev/null
    echo 'echo $SSH_PASSPHRASE' >~/.ssh_askpass && chmod +x ~/.ssh_askpass
    echo "$SSH_PRIVATE_KEY" | tr -d '\r' | DISPLAY=None SSH_ASKPASS=~/.ssh_askpass ssh-add - >/dev/null
```

Oops, forgot to mark executable!

Example GitHub Action Step

What if you want to reuse this across workflows or projects?



A photograph of a person sitting alone on a dark, polished concrete floor against a large, textured concrete wall. The person is wearing a red and black striped beanie, a dark long-sleeved shirt, and jeans. They are sitting cross-legged with their head down, arms wrapped around their knees, in a posture of distress or deep thought. A single electrical outlet with two white covers is visible on the wall to the left. The lighting is dramatic, coming from the side to create strong shadows and highlights on the concrete surfaces.

Me, when maintaining complex workflows



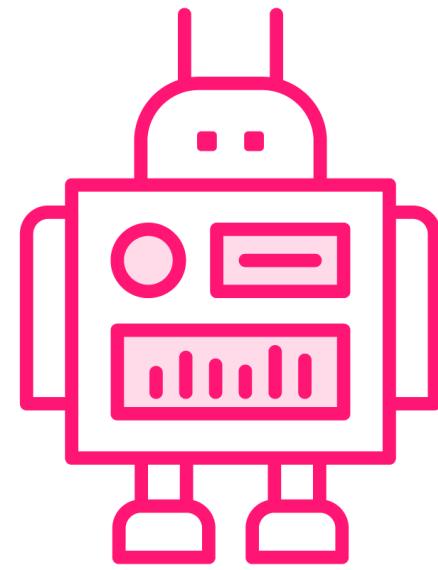
```
- name: Setup SSH passphrase
  env:
    SSH_PASSPHRASE: ${{secrets.VM_SSH_PASSPHRASE}}
    SSH_PRIVATE_KEY: ${{secrets.VM_SSH_KEY}}
  run: |
    ssh-agent -a $SSH_AUTH_SOCK >/dev/null
    echo 'echo $SSH_PASSPHRASE' >~/.ssh_askpass && chmod +x ~/.ssh_askpass
    echo "$SSH_PRIVATE_KEY" | tr -d '\r' | DISPLAY=None SSH_ASKPASS=~/ssh_askpass ssh-add - >/dev/null
```

Example GitHub Action Step

Instead of copy-pasting all over, you can build a custom Action to encapsulate it



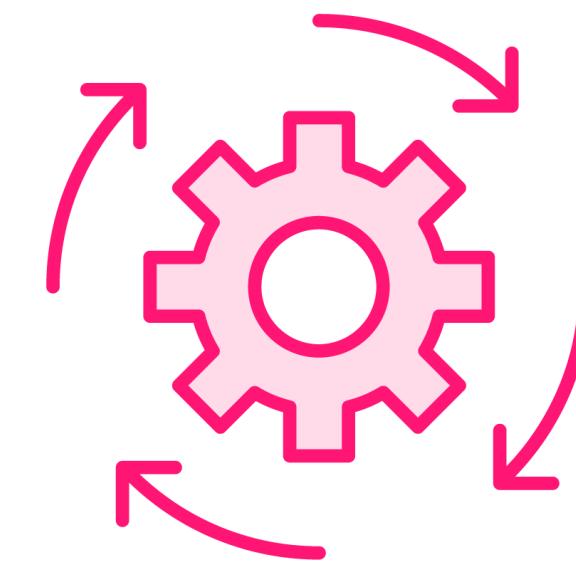
Benefits of Custom GitHub Actions



**Automate and
encapsulate
repetitive tasks**



**Enforce
consistent
processes**



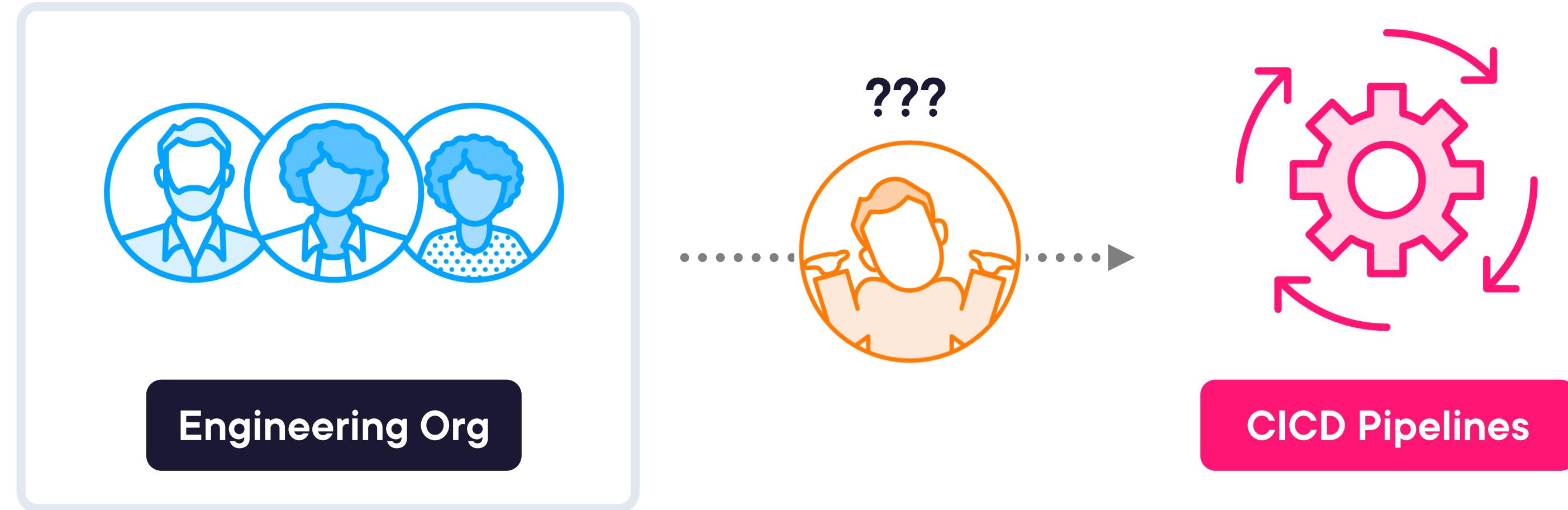
**Boosts
productivity
and quality**



**Improve your
developer
experience**



GOBOMANTICS

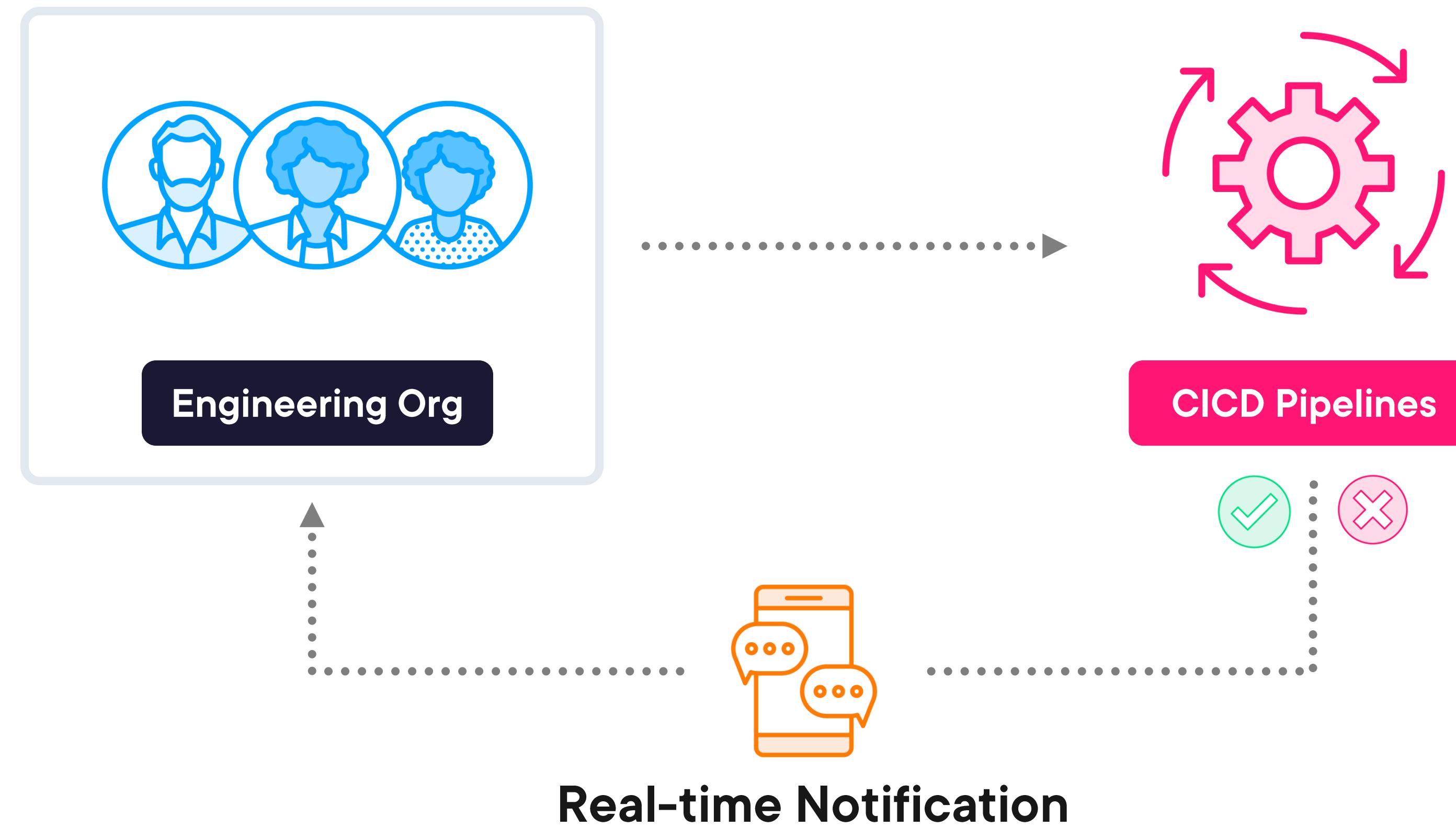




CTO



The Solution: GitHub Actions Status Notifier



Demo: Slack Send Action

GitHub Action

slack-send

v1.26.0 [Latest version](#)

[Use latest version](#)

Slack Send GitHub Action

 codecov 97%

Send data into Slack using this GitHub Action!

 Verified creator
GitHub has verified that this action was created by [slackapi](#).
[Learn more about verified Actions.](#)



The Problem: Associating GitHub Users to Slack Users

notify-author.yaml

```
- name: Notify author directly
  id: slack
  uses: slackapi/slack-github-action@v1
  with:
    channel-id: 'SLACK_USER_ID'
  env:
    SLACK_BOT_TOKEN: ${{ secrets.SLACK_BOT_TOKEN }}
```

How do you associate
the GitHub actor to your
Slack user?



Demo: Slack Message



Pluralsight Demo GitHub Actions APP 2:55 PM

GitHub build result: success

<https://github.com/kamranayub/pluralsight-authoring-github-actions-demo-action/pull/1>

GitHub build result: success

<https://github.com/kamranayub/pluralsight-authoring-github-actions-demo-action/commit/944aafffd318393f843110bfc08b6085d6fac600>



Course Overview

Choosing the right action type

- JavaScript, Docker, or Composite

Authoring the action

- Metadata, syntax, and directory structure
- Coding the action

Troubleshooting the action

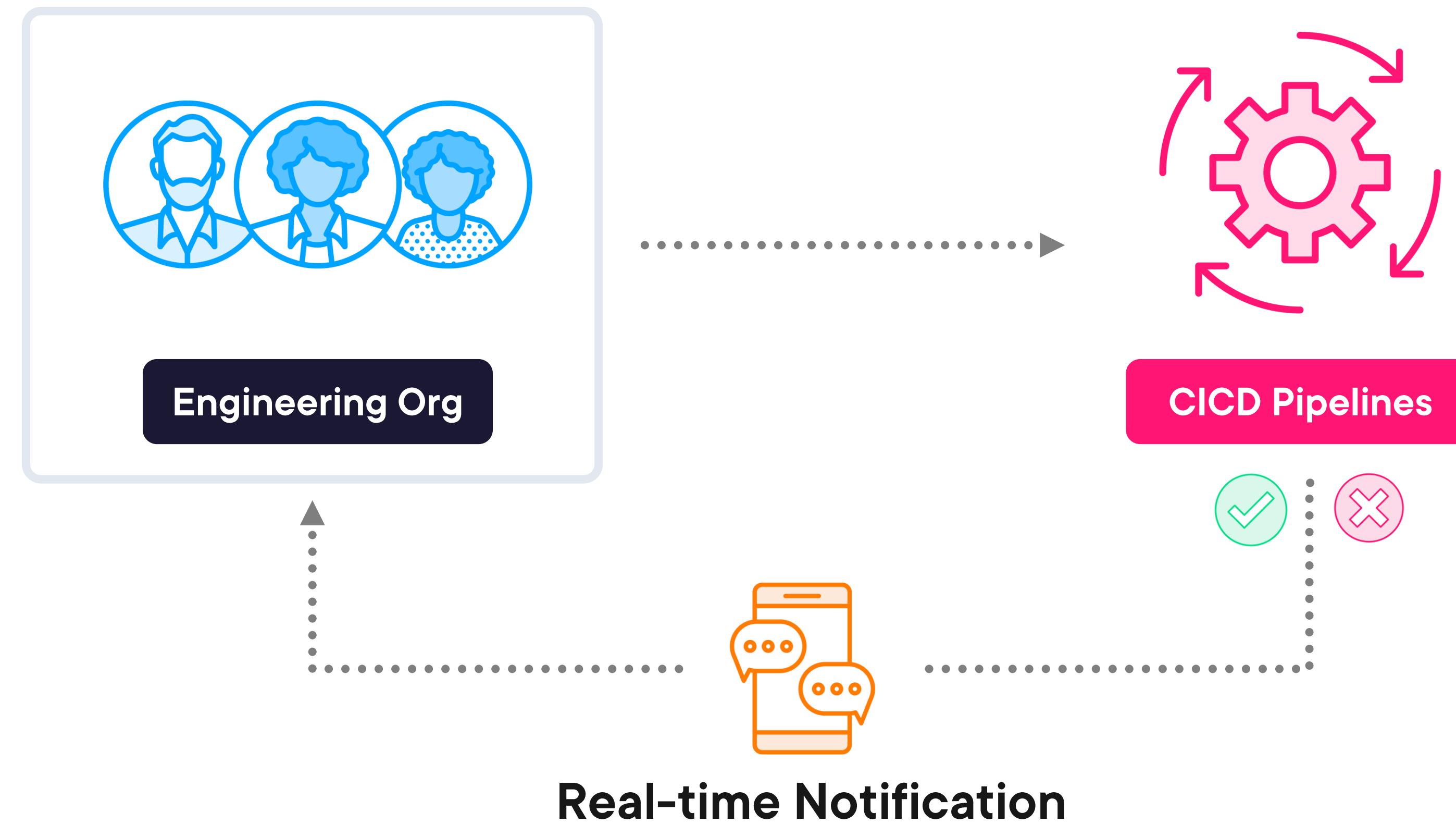
- Logging, debugging, and validation

Publish the action

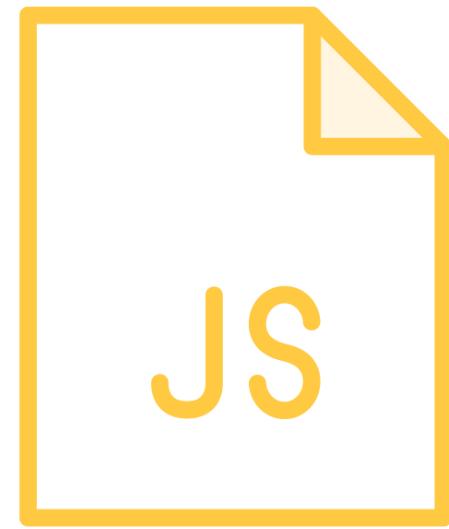
- Versioning and best practices
- Private or public consumption



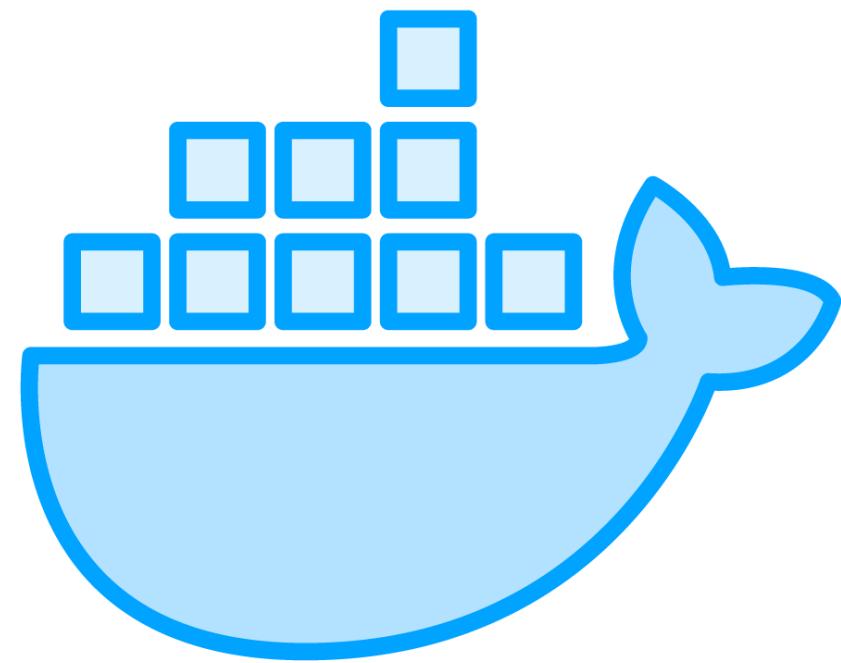
How Do You Make This Real?



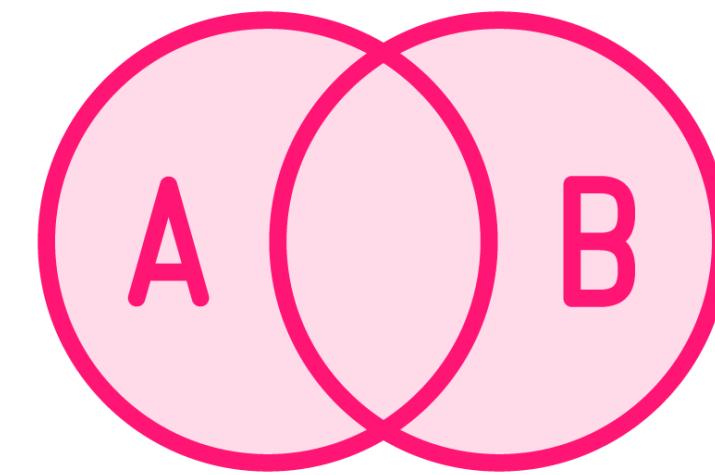
3 Types of GitHub Actions



JavaScript



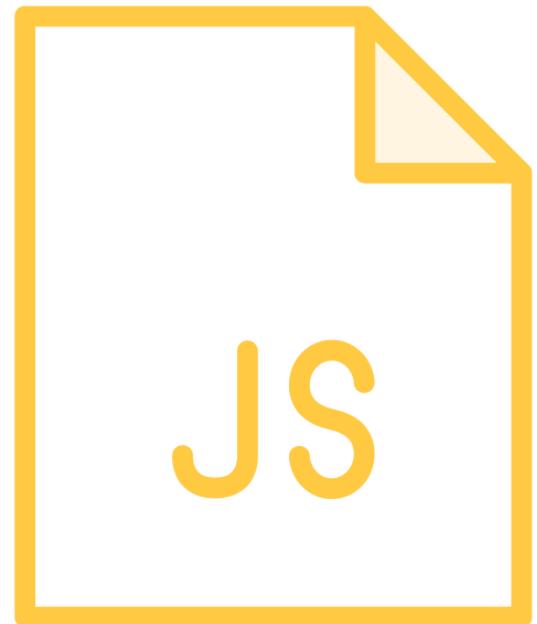
Docker



Composite



JavaScript Actions



Run directly on the runner machine

- Separates action code from the environment

Execute quickly without extra setup

- No need to build or pull down containers

Ideal for quick execution

- Rely on runner's existing environment

Simplifies action code

- Compatible across all GitHub-hosted runners including Ubuntu, Windows, and macOS
- Use the GitHub Actions toolkit for Node.js



JavaScript Actions

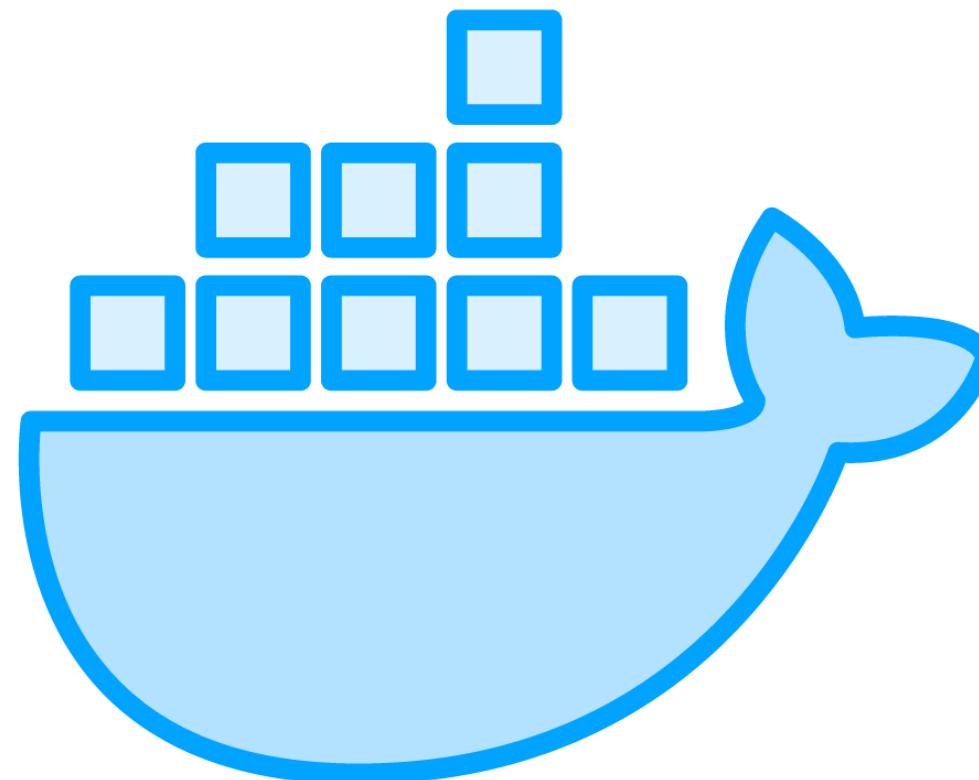
Example YAML schema

action.yml

```
name: 'Demo JavaScript action'  
description: '....'  
runs:  
  using: 'node20'  
  main: index.js
```



Docker Actions



Runs inside a Docker container

- Packages code *with* its environment

Consistent and reliable

- Control the exact environment in which your action runs

Ideal for specialized actions



Docker Actions

Example YAML schema

action.yml

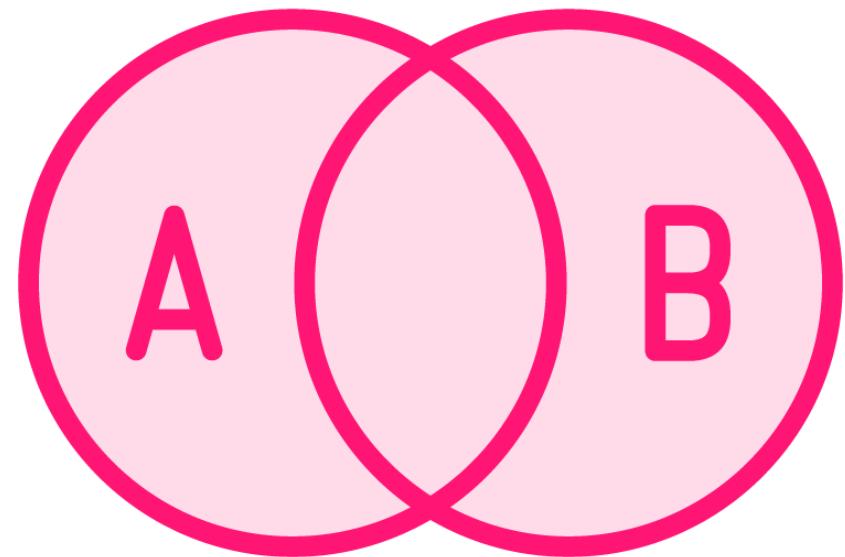
```
name: 'Demo Docker action'  
description: '...'  
runs:  
  using: 'docker'  
  image: 'Dockerfile'  
  args:  
    - arg1  
    - arg2
```

Dockerfile

```
FROM alpine:3.10  
  
COPY entrypoint.sh /entrypoint.sh  
  
ENTRYPOINT [ "/entrypoint.sh" ]
```



Composite Actions



Allows you to use other actions

- Combine multiple actions together

Supports private and public actions

- Reference relative folders in the same repo
- Reference public actions

Ideal for packaging workflows than run multiple actions together



Composite Actions

Example YAML schema

action.yml

```
name: 'Demo Composite action'
description: '...'
runs:
  using: 'composite'
  steps:
    - uses: './get-channel'
      id: get-channel

    - uses: slackapi/slack-github-action@v1.26.0
      with:
        channel-id: ${{ steps.get-channel.outputs.channel-id }}
```

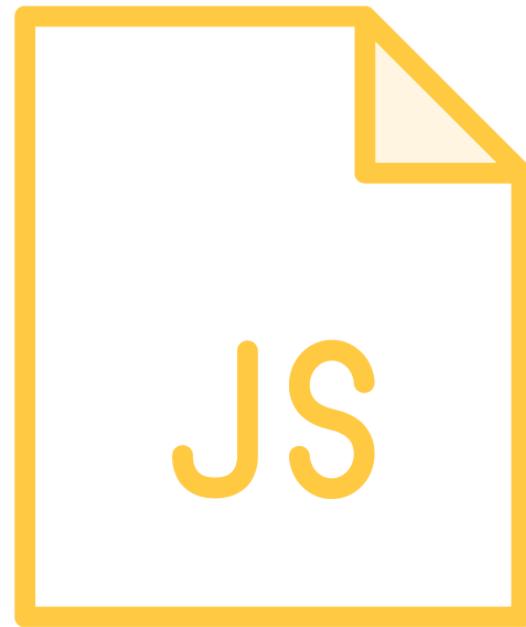


Trade-offs with JavaScript and Docker Actions

JavaScript	vs.	Docker
Just code		Packages environment and code
Simple setup		Can basically accomplish anything
Supported on all runners		Not supported on Windows-based runners
Can't handle specialized tasks such as using graphics (e.g. Chromium)		Requires local Docker tooling to contribute and build



Consider JavaScript Actions When...



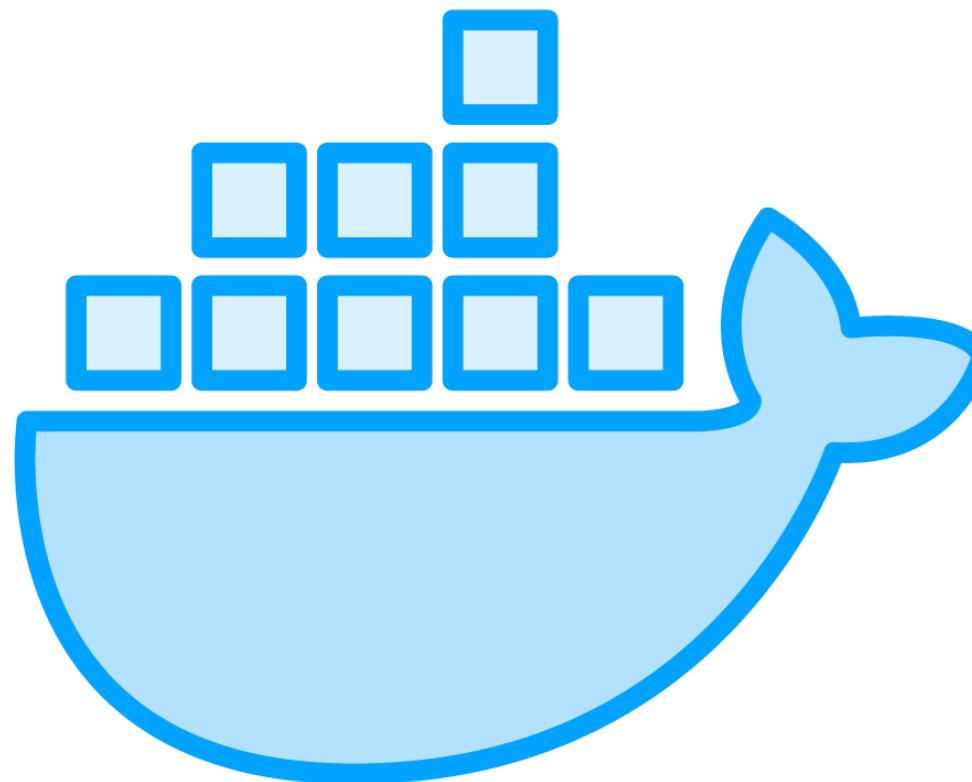
You want to move fast with less maintenance

Everything you need is available in Node.js

Your team is already familiar with JavaScript



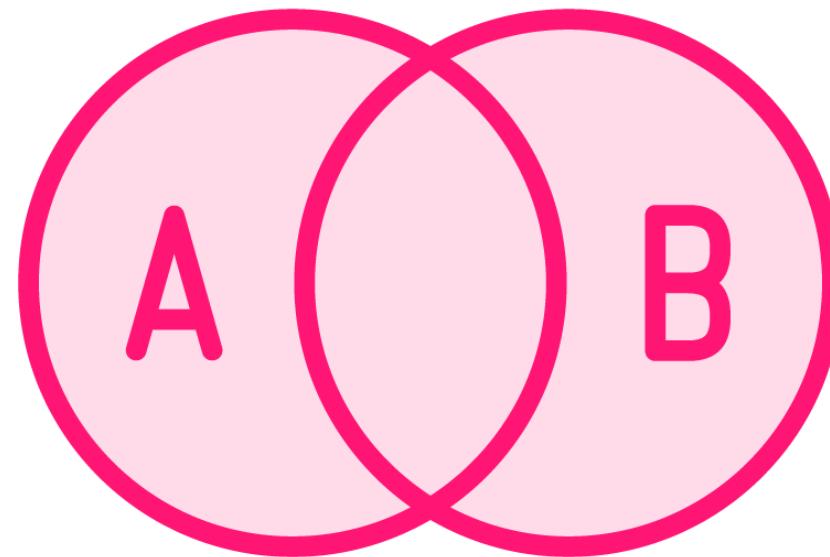
Consider Docker Actions When...



- You need special environment setup like private networking or a database**
- Your team uses a different tech stack**
- You want to wrap existing code**
- You already have pre-built Docker images**



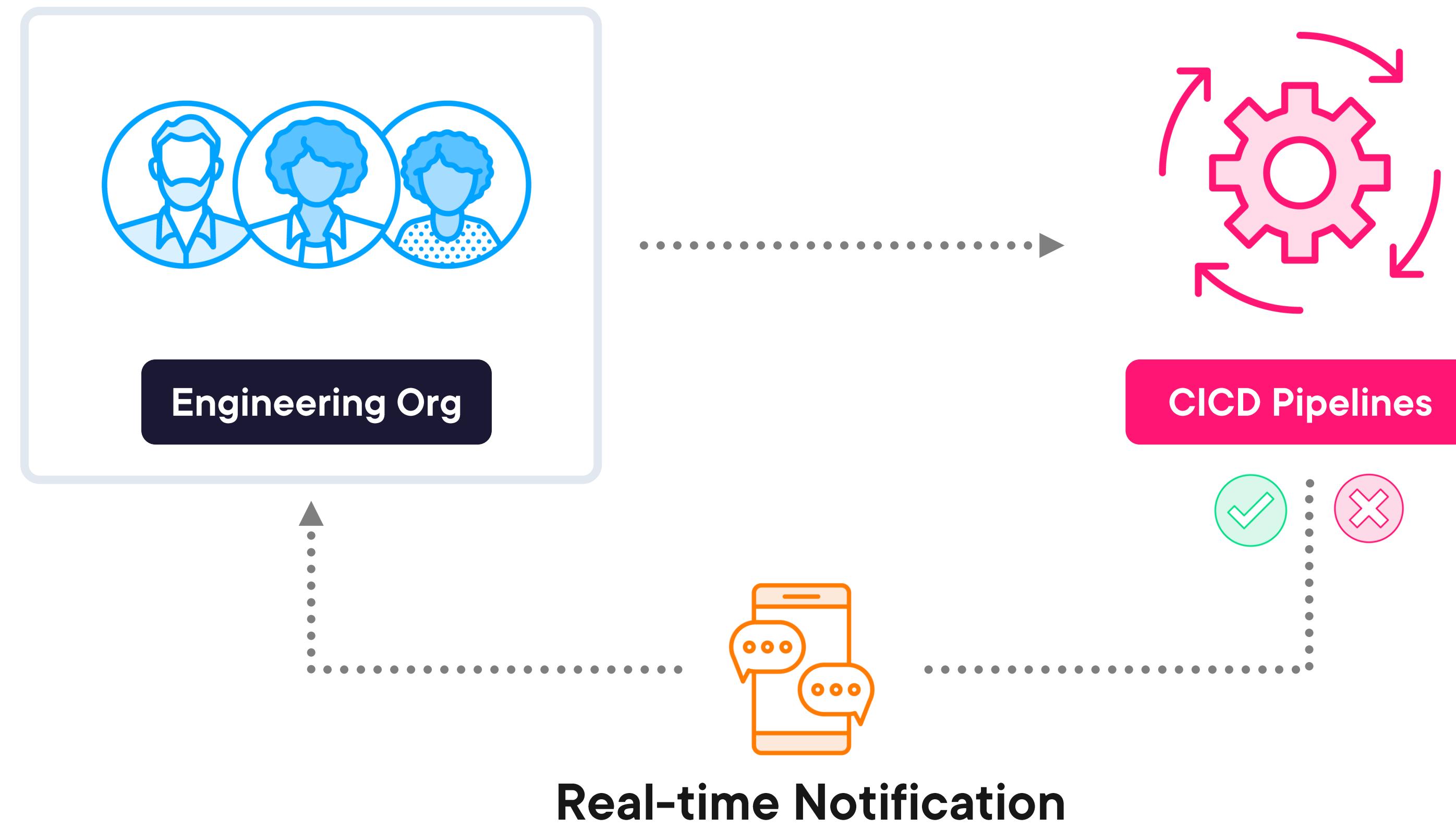
When Should You Use a Composite Action?



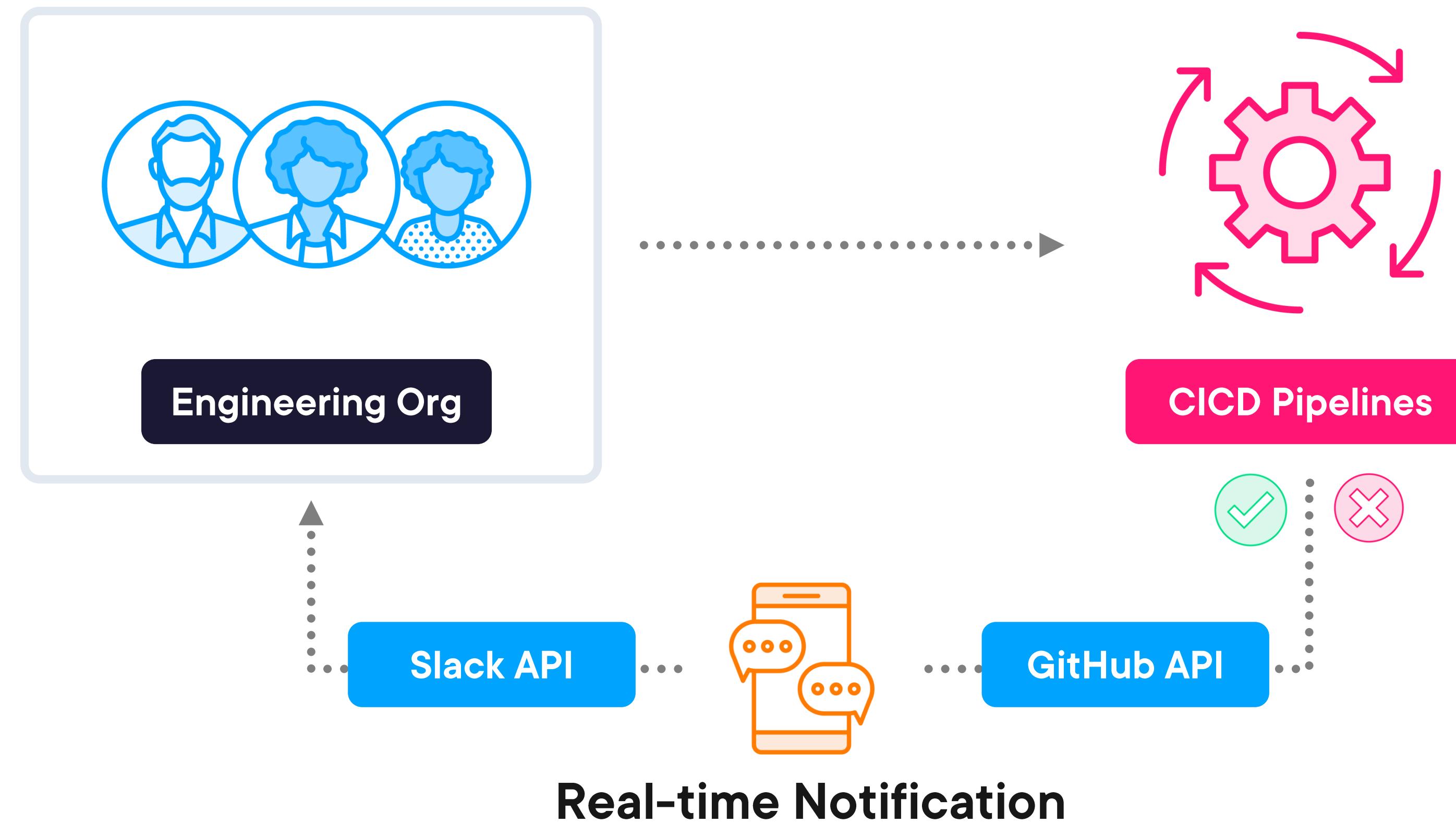
When you need to encapsulate an end-to-end workflow comprising of multiple public or private actions



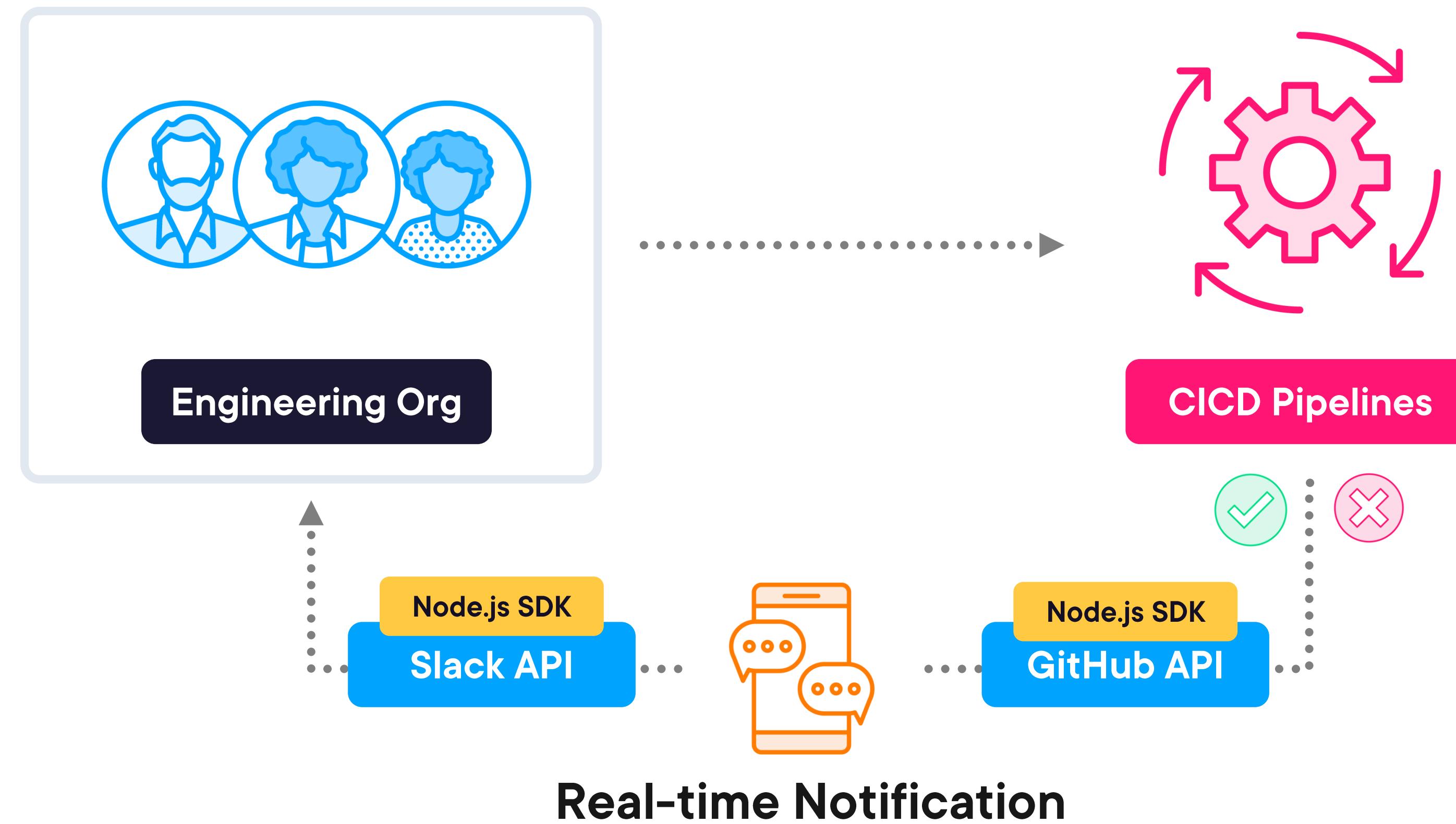
Applying to CI/CD Pipeline Status Notifier



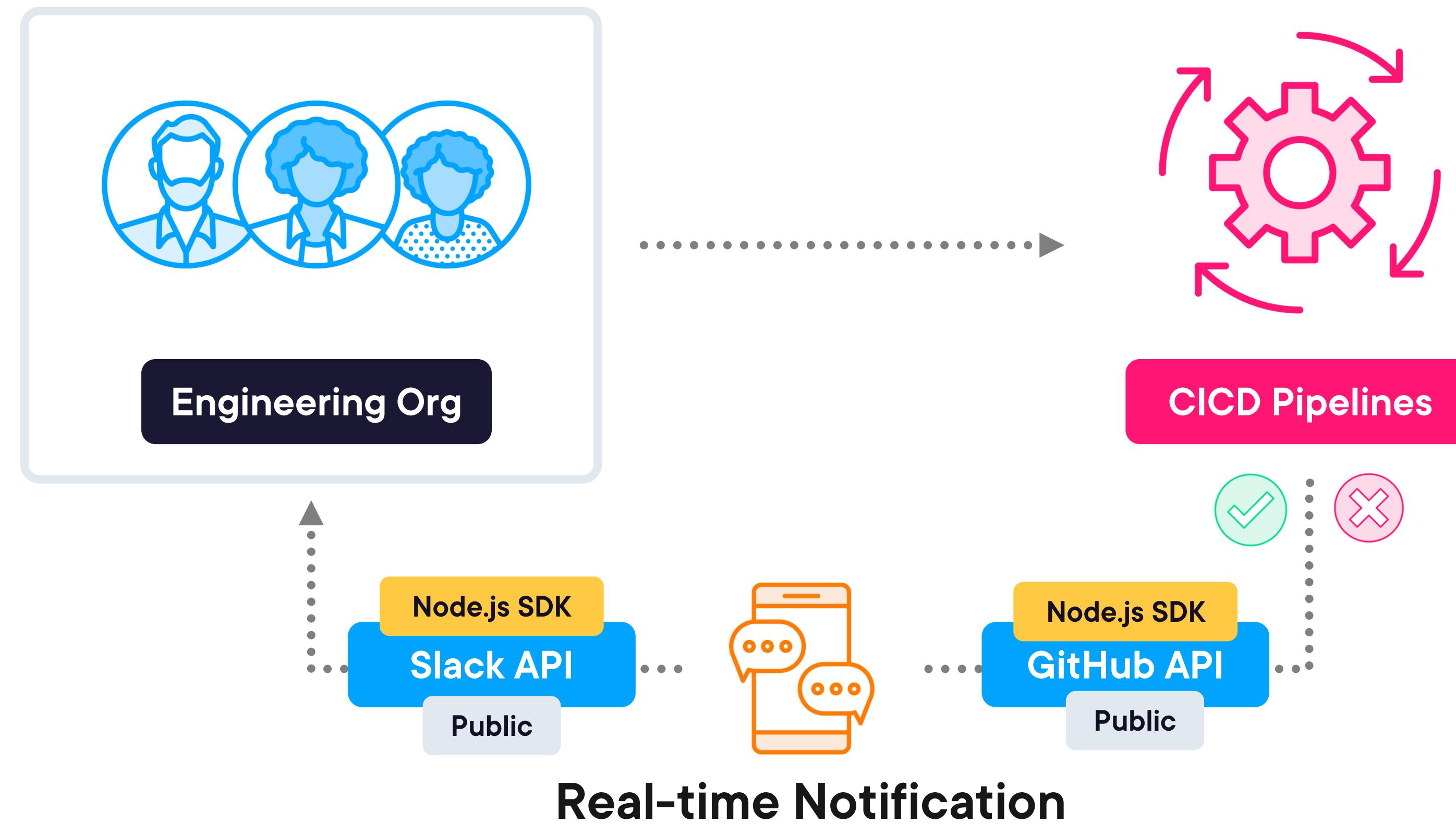
Applying to CI/CD Pipeline Status Notifier



Applying to CI/CD Pipeline Status Notifier



Applying to CI/CD Pipeline Status Notifier



Applying to CI/CD Pipeline Status Notifier



Look up email address of GitHub actor ([GitHub API](#))



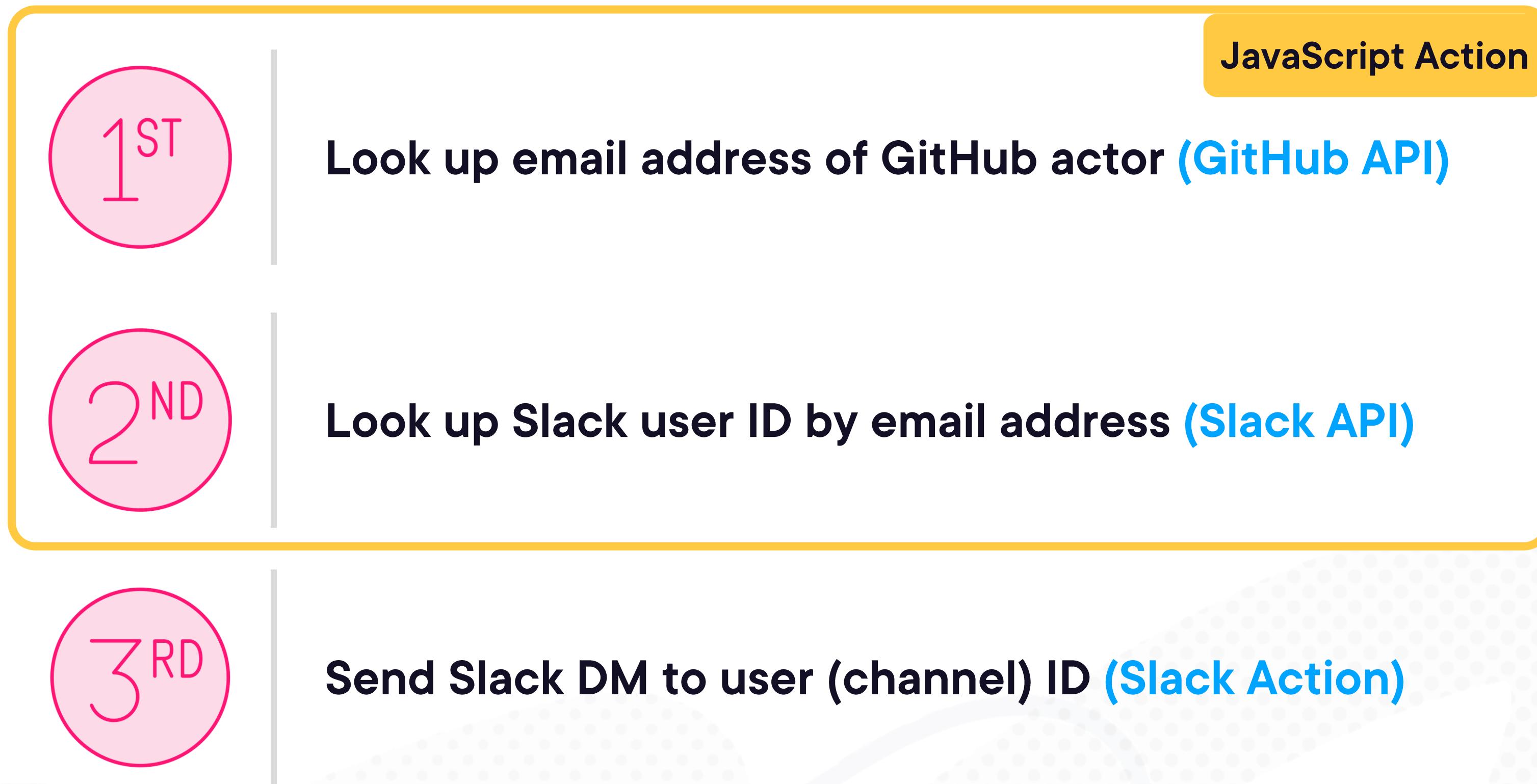
Look up Slack user ID by email address ([Slack API](#))



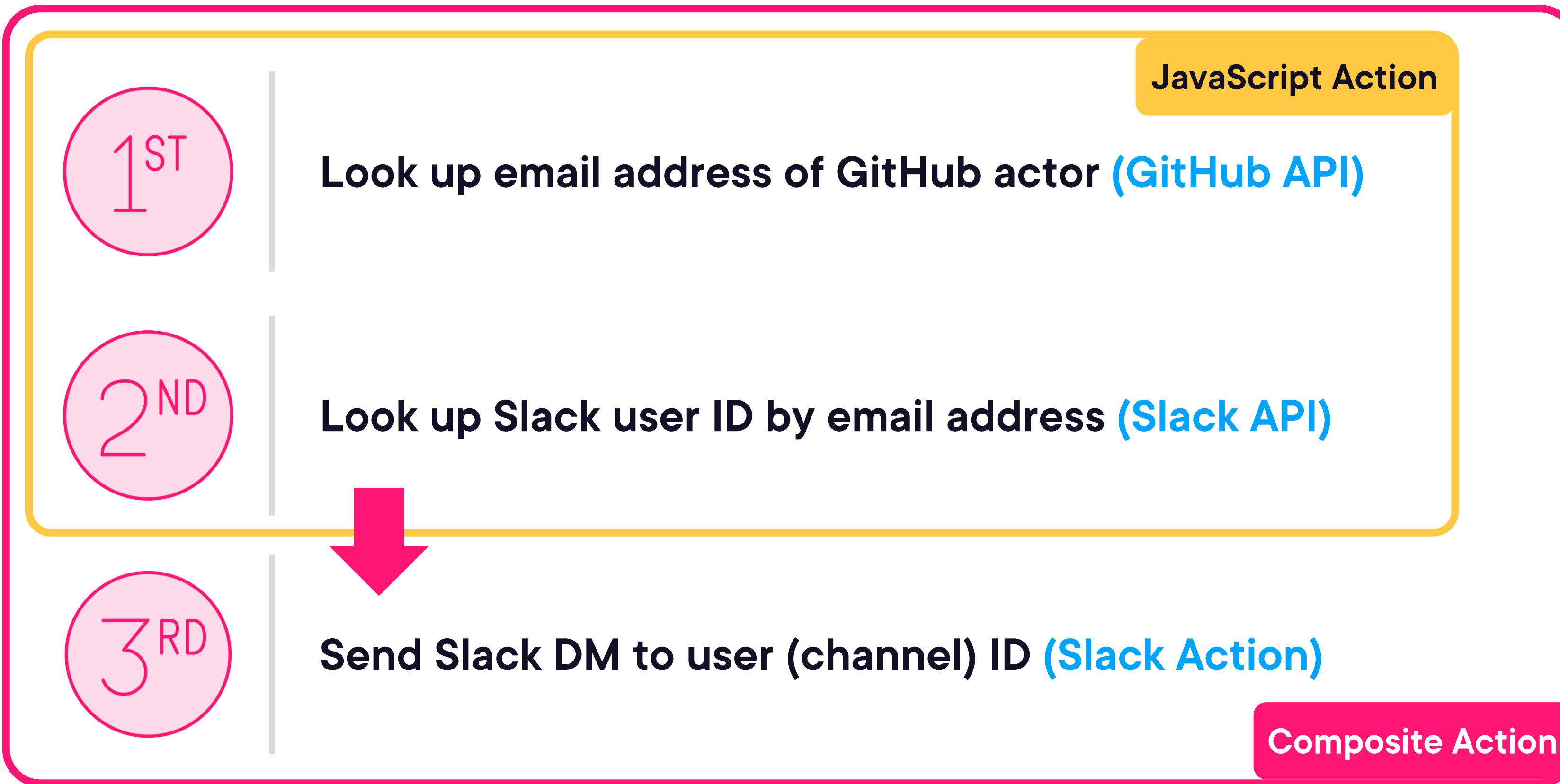
Send Slack DM to user (channel) ID ([Slack Action](#))



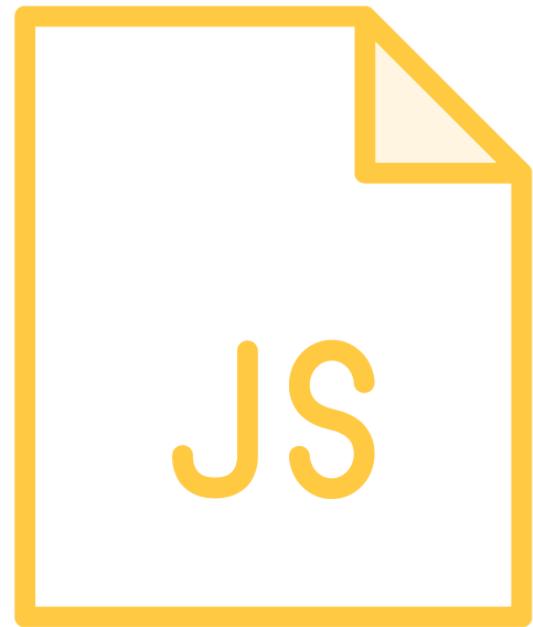
Applying to CI/CD Pipeline Status Notifier



Applying to CI/CD Pipeline Status Notifier



Tooling for JavaScript Actions

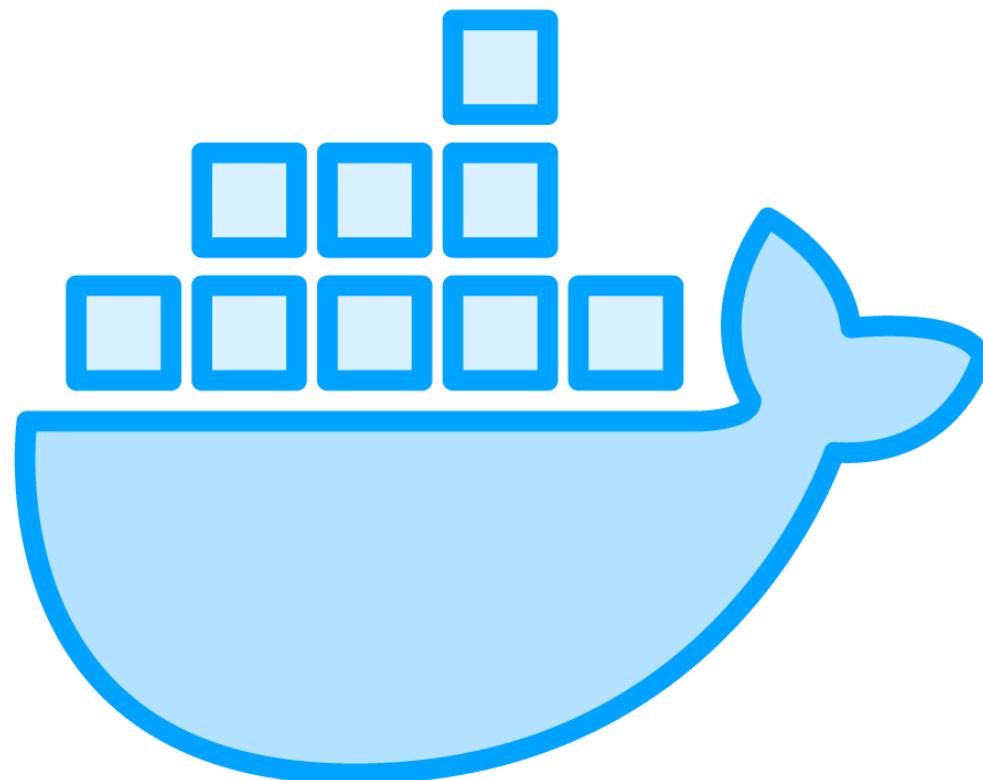


Git

Node.js and npm



Tooling for Docker Actions



Git

Tooling for your tech stack

- Python, Ruby, .NET, etc.

Docker Tooling

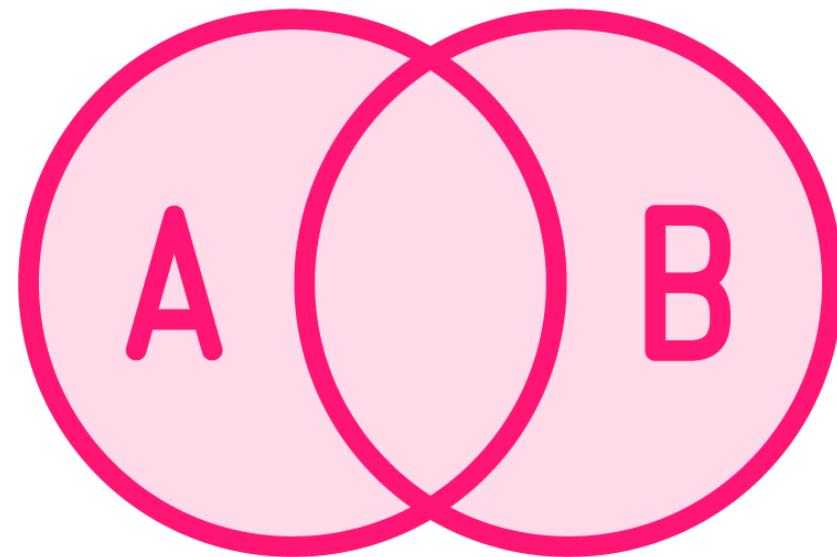
- If running locally
- e.g. Docker for Desktop

Windows Subsystem for Linux (WSL2)

- AKA "Bash on Windows"



Tooling for Composite Actions



Git

Tooling for other action types

- If you're also building custom actions to consume in the composite action



Installation References

Windows Subsystem for Linux

<https://learn.microsoft.com/en-us/windows/wsl/about>

Docker for Desktop

<https://www.docker.com/>

Node.js

<https://www.nodejs.org>

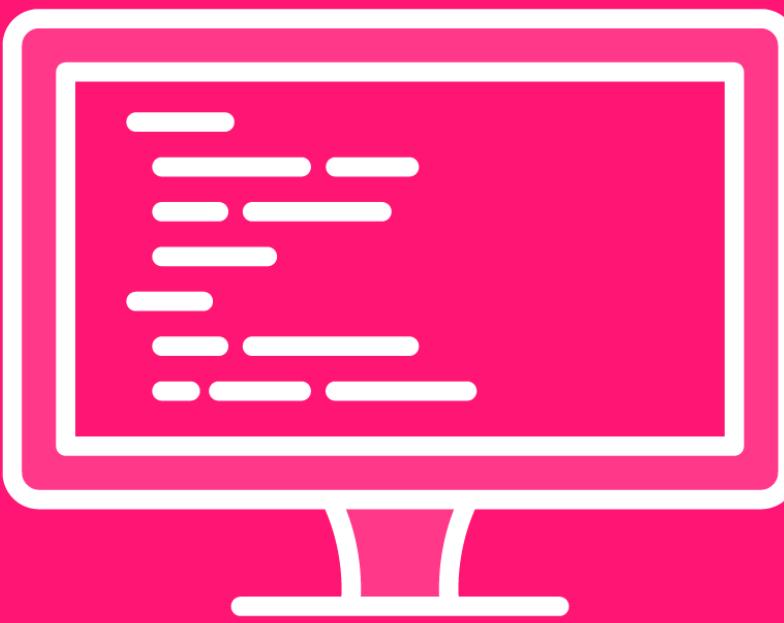


Demo: VS Code

```
pluralsight-authoring-github-actions-demo-action [* 72.84°F] [main]
● > node -v
v20.12.2
pluralsight-authoring-github-actions-demo-action [cloud 84.65°F] [main]
● > npm -v
10.5.0
pluralsight-authoring-github-actions-demo-action [cloud 84.65°F] [main]
● > docker -v
Docker version 26.1.1, build 4cf5afa
pluralsight-authoring-github-actions-demo-action [cloud 84.65°F] [main]
> █
```

```
pluralsight-authoring-github-actions-demo-action [cloud 84.65°F] [main]
● > git --version
git version 2.34.1.windows.1
```





github.com/kamranayub/pluralsight-course-authoring-maintaining-github-actions

**Find the demo files in the Exercise Files of the
course or this companion GitHub repository**



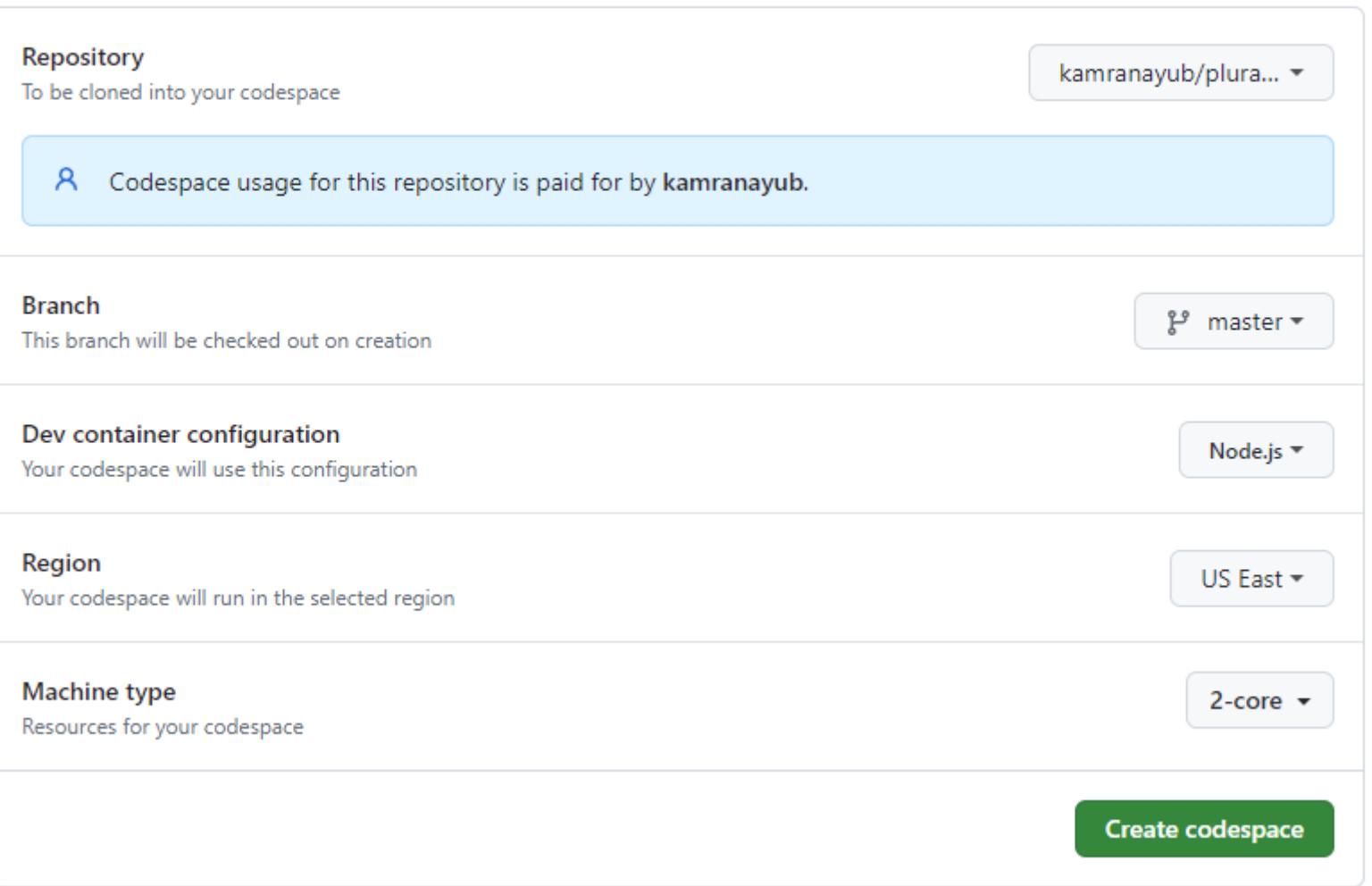
GitHub Codespaces

Follow Along

 Open in GitHub Codespaces

You can follow along with the demos in the course on your own using the provided GitHub Codespace devcontainer. Everything is pre-configured for you, so you don't need to install anything on your local machine. All GitHub Free and Pro accounts have a [free monthly quota](#) for Codespace usage.

Create a new codespace



The screenshot shows the 'Create a new codespace' form on GitHub. It consists of five sections with dropdown menus:

- Repository:** Set to "kamranayub/plura...".
Description: "To be cloned into your codespace".
- Branch:** Set to "master".
Description: "This branch will be checked out on creation".
- Dev container configuration:** Set to "Node.js".
Description: "Your codespace will use this configuration".
- Region:** Set to "US East".
Description: "Your codespace will run in the selected region".
- Machine type:** Set to "2-core".
Description: "Resources for your codespace".

A large green "Create codespace" button is at the bottom right.





Building a JavaScript Development Environment

Kamran Ayub

