BuildKite: In *My* Daemon Node?

M. Cuffaro

A

$$GH \rightleftarrows BK \rightleftarrows HPG$$

Problem:

> ⚠ **Problem**
>
> Our builds take a long time!

> ℹ **But…**
>
> We have access to a HPC

> 💡 **Solution: Buildkite!**
>
> ▶ CI/CD service trusted by CLiMA lab.
> ▶ BK can watch our github repositories for changes.
> ▶ When triggered, BK can execute scripts (such as SLURM)

How does BuildKite work?

> **i** Gist
>
> **sources** send POST requests to **agents** which execute **pipelines** consisting of **steps**.
>
> $$\text{source} \xrightarrow{\text{POST}} \text{agent} \xrightarrow{\text{kicks off}} \text{pipeline}$$
>
> pipeline defined in `repo/.buildkite/pipeline.yml`

> **i** How does the source know where to POST?
>
> The repo's **webhook** specifies *where* to send the payload. The webhook is a function of the agent

Problem: How does BK mediate GH and HPG?

> 💡 **Daemons**
>
>
>
> Figure 1: Daemon
>
> ▶ *Daemons* are continuous background processes.
> ▶ HPC tolerates daemons with small footprints.
> ▶ We will host BuildKite agent as a daemon on this node

Architecture

1. Set-up: Agent

An agent is installed on the daemon node and started. It lies dormant…

2. Set-up: Install BuildKite onto Github

▶ The Github application allows repos to configure **webhooks**.
▶ During any of the subscribed events, Github sends a POST request to **payload URL**
▶ The daemon exposes an API to Github, waiting f

3. Set-up: Create Pipeline

▶ Specify repository

4. Set-up: Webhook Manually

Go to pipeline dashboard > set up webhook

Creating Pipeline

Pipelines are workflows executed by agents. They are specified

Repo

The agent receives the POST request and executes its steps:

```
steps:
  - label: ":pipeline: Pipeline upload"
  - command: buildkite-agent pipeline upload
```

This looks at the `.buildkite` folder.

```
steps:
  - label: ":sunrise: wake up!"
    command: echo "the sky is awake, so I'm awake, so we ha
```