GitHub Actions for Scientific Data Workflows

***Valentina Staneva**, Senior Data Scientist*

eScience Institute, Paul G. Allen School of Computer Science & Engineering

University of Washington

Oct. 2, 2023

US
RSE
2023

eScience Institute
ADVANCING DATA-INTENSIVE DISCOVERY IN ALL FIELDS

Today’s Plan

Schedule

- Overview of Github Actions and Workflows (15 min)
- Setting up your first workflow: a scientific Python environment (15 min)
- Scheduled algorithm deployment to a real-time stream (20 min)
- Exporting results (20 min)
- More workflow ideas (20 min)

Slides: https://docs.google.com/presentation/d/1nf8QUO7YtbJj-ZUclYTbaPnP-snKqZxLUQhTKLWxt5k

Github Repository: https://github.com/valentina-s/GithubActionsTutorial-RSE23

Learning Objectives

- Learners distinguish between Github Actions and Workflows and understand their role within the software development cycle
- Learners are capable of triggering GitHub Action Workflows in several different ways and can determine which method could be useful in typical data science applications
- Learners can export (data) outputs of Github Action Workflows, e.g. tables, plots.

Leave with your own ideas on how to integrate Github Actions in your own work!

What are GitHub Actions?

GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline.

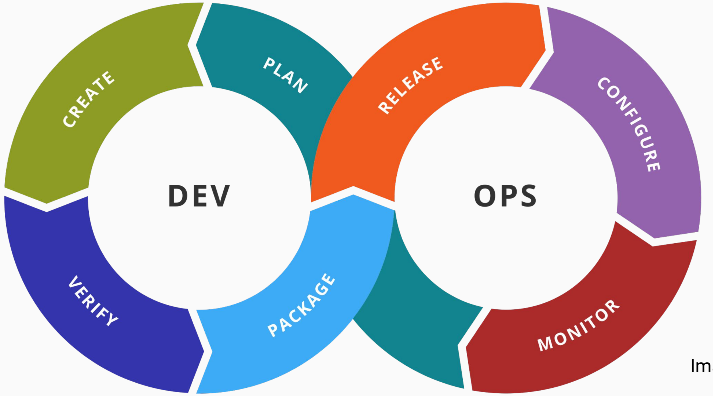
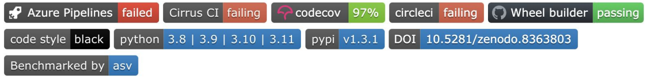


Image Source: Wikipedia

Automatic:

- ➢ Style Checking
- ➢ Testing
- ➢ Coverage Report Generation
- ➢ Documentation Building
- ➢ Package Building
- ➢ Publishing (to PiPy)

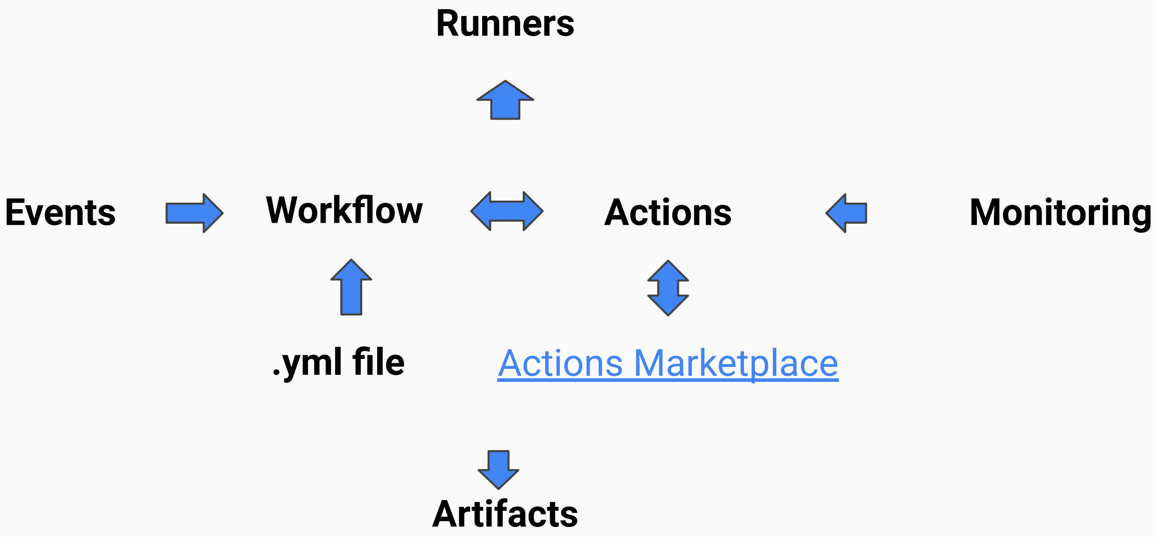


https://github.com/scikit-learn/scikit-learn

What More are GitHub Actions?

A platform to run any (not too complex) workflow in a virtual environment and integrate with GitHub.

Github Actions Platform



Runners

Events

Workflow

Actions

Monitoring

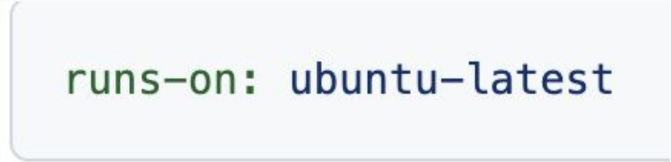
.yml file

[Actions Marketplace]

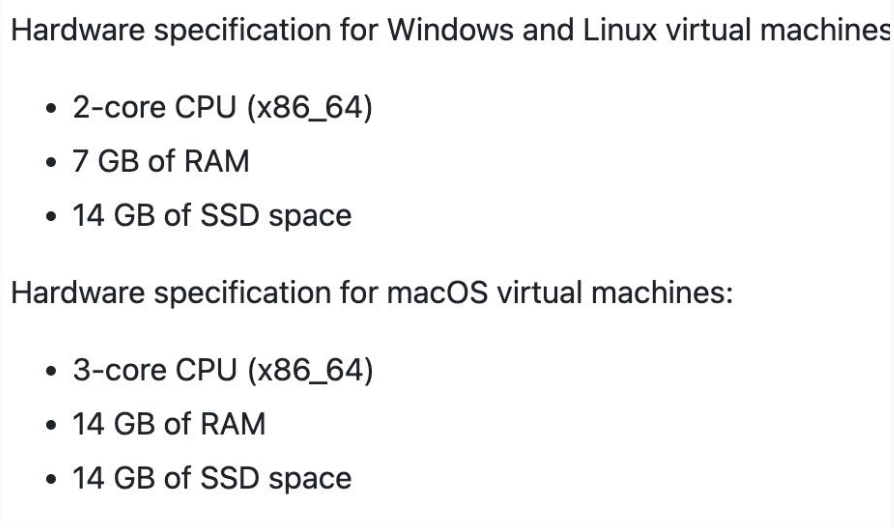
Artifacts

Runners: Virtual Computing Environment

- Github-Hosted Runners (Free!)
  - ➢ Ubuntu
  - ➢ MacOS
  - ➢ Windows Server



```
runs-on: ubuntu-latest
```



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:

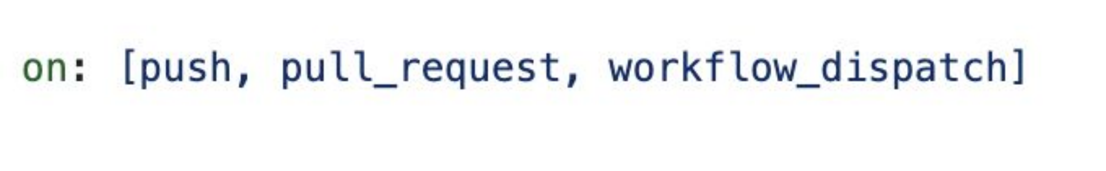
- 3-core CPU (x86_64)
- 14 GB of RAM
- 14 GB of SSD space

- Large runners: Github Enterprise Cloud

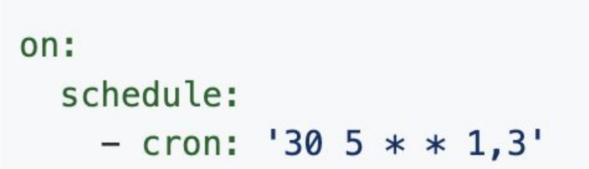
- Self-Hosted Runners

Trigger Events

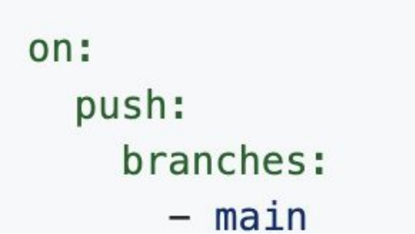
- Events that occur in your workflow's repository
- Events that occur outside of GitHub and trigger a `repository_dispatch` event on GitHub
- Scheduled times
- Manual



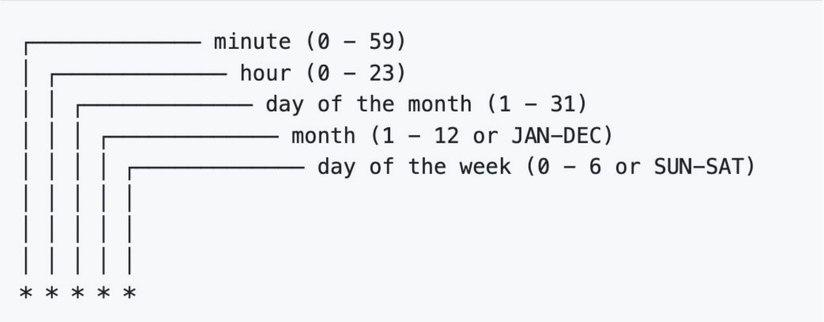
```
on: [push, pull_request, workflow_dispatch]
```



```
on:
  schedule:
    - cron: '30 5 * * 1,3'
```

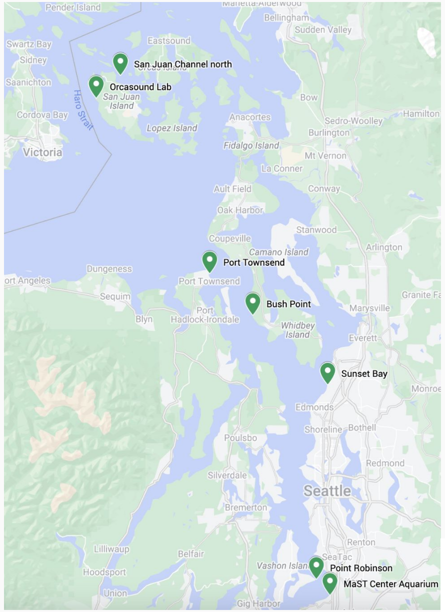


```
on:
  push:
    branches:
      - main
```



```
┌───────────── minute (0 - 59)
│ ┌───────────── hour (0 - 23)
│ │ ┌───────────── day of the month (1 - 31)
│ │ │ ┌───────────── month (1 - 12 or JAN-DEC)
│ │ │ │ ┌───────────── day of the week (0 - 6 or SUN-SAT)
│ │ │ │ │
│ │ │ │ │
│ │ │ │ │
│ │ │ │ │
* * * * *
```

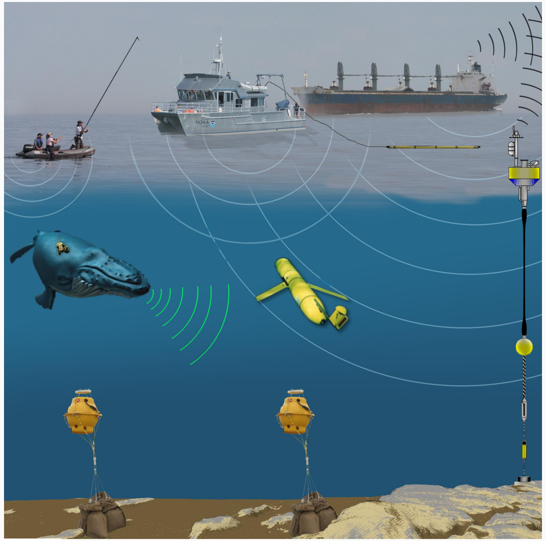
Orcasound: Hydrophone Network and Open Source Community

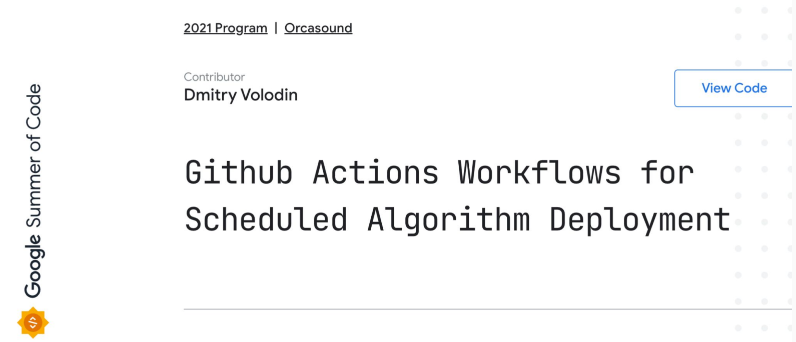


ORCASOUND

Listen for whales

LISTEN LIVE





2021 Program | Orcasound

Contributor
Dmitry Volodin

View Code

Github Actions Workflows for Scheduled Algorithm Deployment

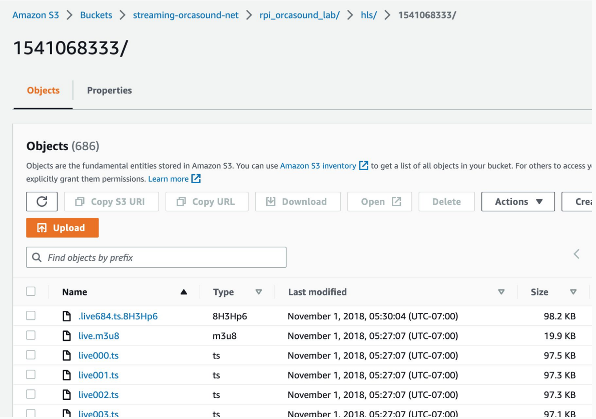
Scientific Data Workflow Example

1. Access data
2. Process
3. Visualize

10 sec segments on AWS S3

`.ts -> .wav`

Spectrogram



Amazon S3 > Buckets > streaming-orcasound-net > rpi_orcasound_lab/ > hls/ > 1541068333/

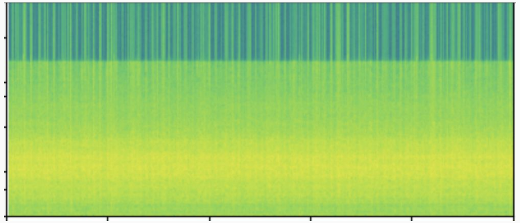
1541068333/

Objects (686)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access y
explicitly grant them permissions. Learn more

	Name	Type	Last modified	Size
□	.live684.ts.8H3Hp6	8H3Hp6	November 1, 2018, 05:30:04 (UTC-07:00)	98.2 KB
□	live.m3u8	m3u8	November 1, 2018, 05:27:07 (UTC-07:00)	19.9 KB
□	live000.ts	ts	November 1, 2018, 05:27:07 (UTC-07:00)	97.5 KB
□	live001.ts	ts	November 1, 2018, 05:27:07 (UTC-07:00)	97.3 KB
□	live002.ts	ts	November 1, 2018, 05:27:07 (UTC-07:00)	97.3 KB
□	live003.ts	ts	November 1, 2018, 05:27:07 (UTC-07:00)	97.1 KB

.wav



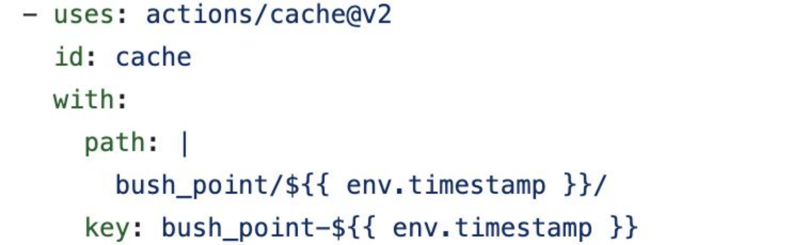
Workflow Steps

- Set up environment
    - Python
    - Scientific packages
- Set date to environment variable
- Set cache path
- Script to generate spectrogram from a file (.png)
    - Download files from S3
    - Convert ts format (not popular) to wav, create spectrogram from wav, save png

Let's Get Started!

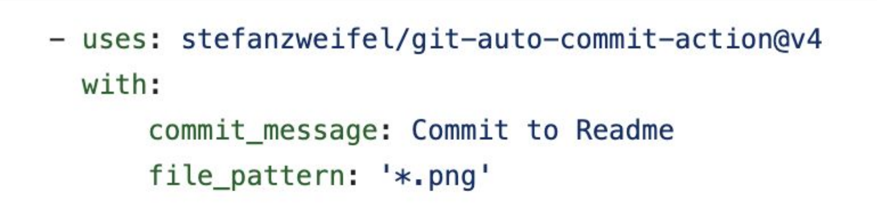
Storing Results

- **Caching**



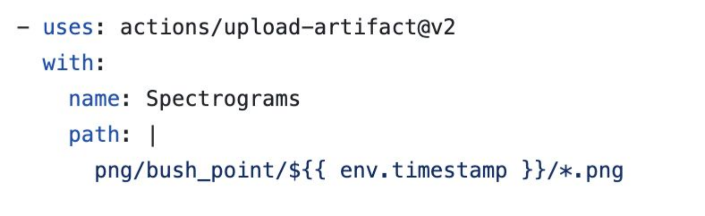
```
- uses: actions/cache@v2
  id: cache
  with:
    path: |
      bush_point/${{ env.timestamp }}/
    key: bush_point-${{ env.timestamp }}
```

- **Committing to GitHub**



```
- uses: stefanzweifel/git-auto-commit-action@v4
  with:
      commit_message: Commit to Readme
      file_pattern: '*.png'
```

- **Artifacts**



```
- uses: actions/upload-artifact@v2
  with:
    name: Spectrograms
    path: |
      png/bush_point/${{ env.timestamp }}/*.png
```

- **Uploading to own storage**
  - Cloud storage
  - Google Drive
  - ...

Creating Your Own GitHub Actions

- ➢ Create a repository for the action
- ➢ Create a `Dockerfile` to run the
- ➢ Create an `action.yml` file to configure the action
- ➢ Create an `entrypoint.sh` with the action steps

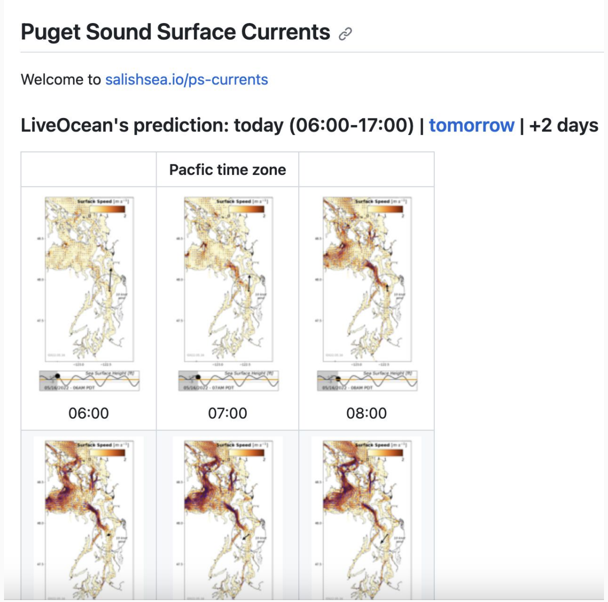
https://docs.github.com/en/actions/creating-actions/creating-a-docker-container-action

https://github.com/actions/hello-world-docker-action

https://github.com/orcasound/get-ooi-data/blob/main/.github/actions/get-ooi-data/action.yml

More Examples

Live Ocean Current Prediction, by Val Veirs



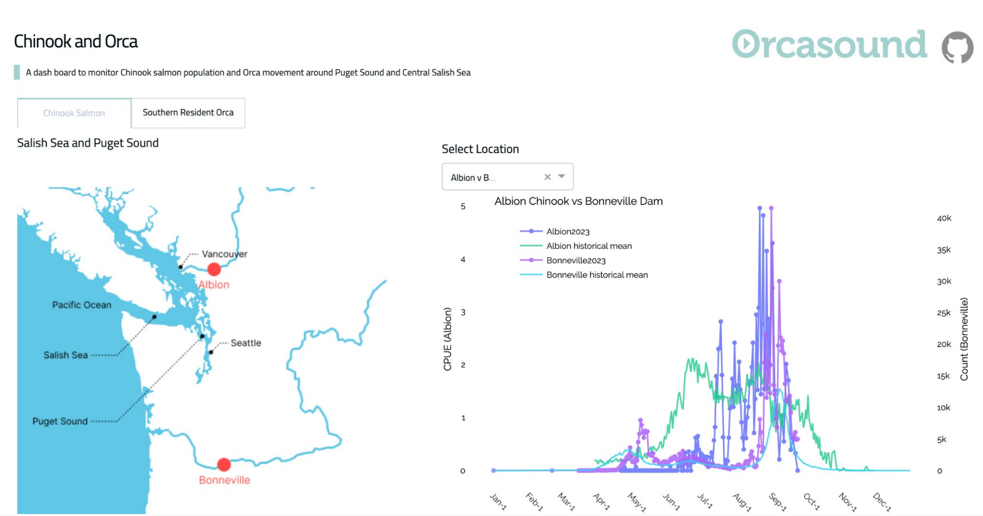
Puget Sound Surface Currents

Welcome to salishsea.io/ps-currents

LiveOcean's prediction: today (06:00-17:00) | tomorrow | +2 days

	Pacfic time zone	
06:00	07:00	08:00

Orca&Salmon Dashboard Data Update, by Zoe Liu



Chinook and Orca

A dash board to monitor Chinook salmon population and Orca movement around Puget Sound and Central Salish Sea

Salish Sea and Puget Sound

Select Location: Albion v B...

Albion Chinook vs Bonneville Dam

https://github.com/salish-sea/ps-currents/tree/main/docs

https://github.com/liu-zoe/orcasalmon/tree/main/.github/workflows

More Ideas

- Data plot is stored in json format and displayed with an interactive library in gh-pages website
- As new data gets updated, check if data is within reasonable scientific bounds
- Different users submit their own version of a model to predict whales in the stream, the model, outcome, user-id, date are stored in a benchmarking table, displayed on readme
- Speed of processing data is recorded, and stored to be visualized in performance plot
- A user submits a new set of training data and that triggers retraining an algorithm, and the output model is stored on github

What ideas do you have how to create Github Actions workflows in your work?