# 🐍 Run Fast!

Catch Performance Regressions in Python

Bencher

# Everett Pompeii

everett@bencher.dev

https://github.com/bencherdev/bencher

Bencher

## How to catch performance regressions?

## Detection ➡ Prevention

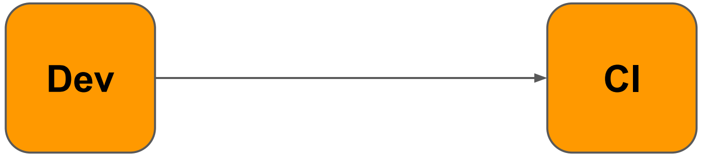
When do performance regression get detected?

Bencher

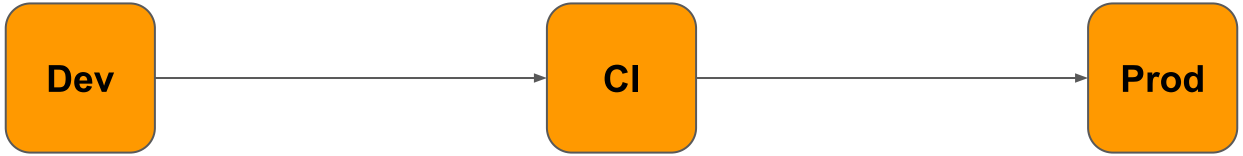
## When do performance regression get detected?

## When do performance regression get detected?

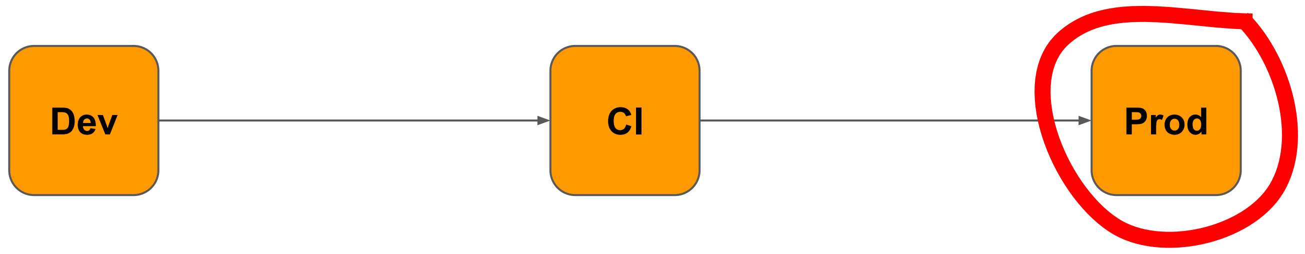
## When do performance regression get detected?



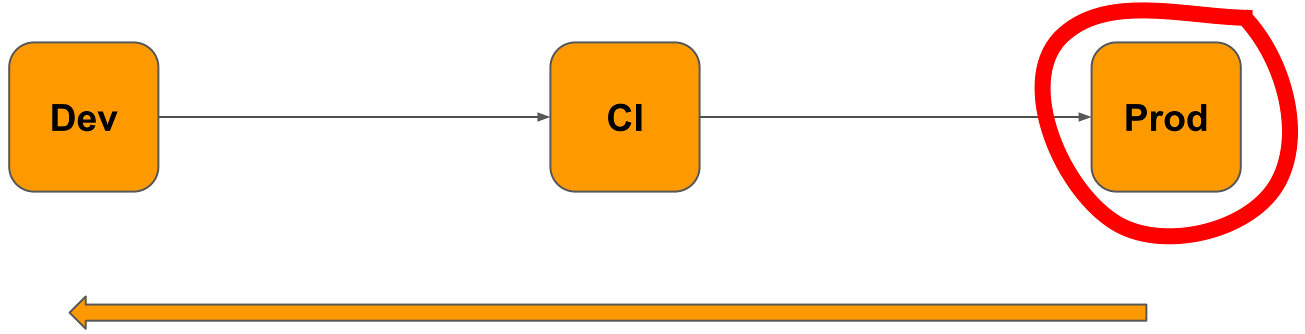
## When do performance regression get detected?



## When do performance regression get detected?



## When do performance regression get detected?



Bencher

## This Is a Work of Fiction

## App v0

## App v0: Calendar App API

Bencher

# App v0: Calendar App API

Bencher

## App v0: Calendar App API

## App v0: Calendar App API

django

FastAPI

## App v0: Calendar App API

## App v0: Calendar App API

Fun Notification Feature

## App v1: Fizz Feature

Bencher

## App v1: Fizz Feature

Return “Fizz” if day is divisible by 3.

## App v1: Fizz Feature

Return “Fizz” if day is divisible by 3.
Otherwise, return None.

## App v1: Fizz Feature

Return “Fizz” if day is divisible by 3.
Otherwise, return None.

```python
def fun_notification(n):
    if not n % 3:
        return 'Fizz'
    return None
```

# App v1: Fizz Feature

1

AUG
2022

| Mo | Tu | We | Th | Fr | Sa | Su |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |

September

| M | T | W | T | F | S | S |
|---|---|---|---|---|---|---|
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |

October

| M | T | W | T | F | S | S |
|---|---|---|---|---|---|---|
| 26 | 27 | 28 | 29 | 30 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 1 | 2 | 3 | 4 | 5 | 6 |

Bencher

## App v1: Calendar App API

Improved Fun Notification Feature

Bencher

## App v2: FizzBuzz Feature

## App v2: FizzBuzz Feature

Return “Fizz” if day is divisible by 3

## App v2: FizzBuzz Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5

## App v2: FizzBuzz Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both.

## App v2: FizzBuzz Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Otherwise, return None.

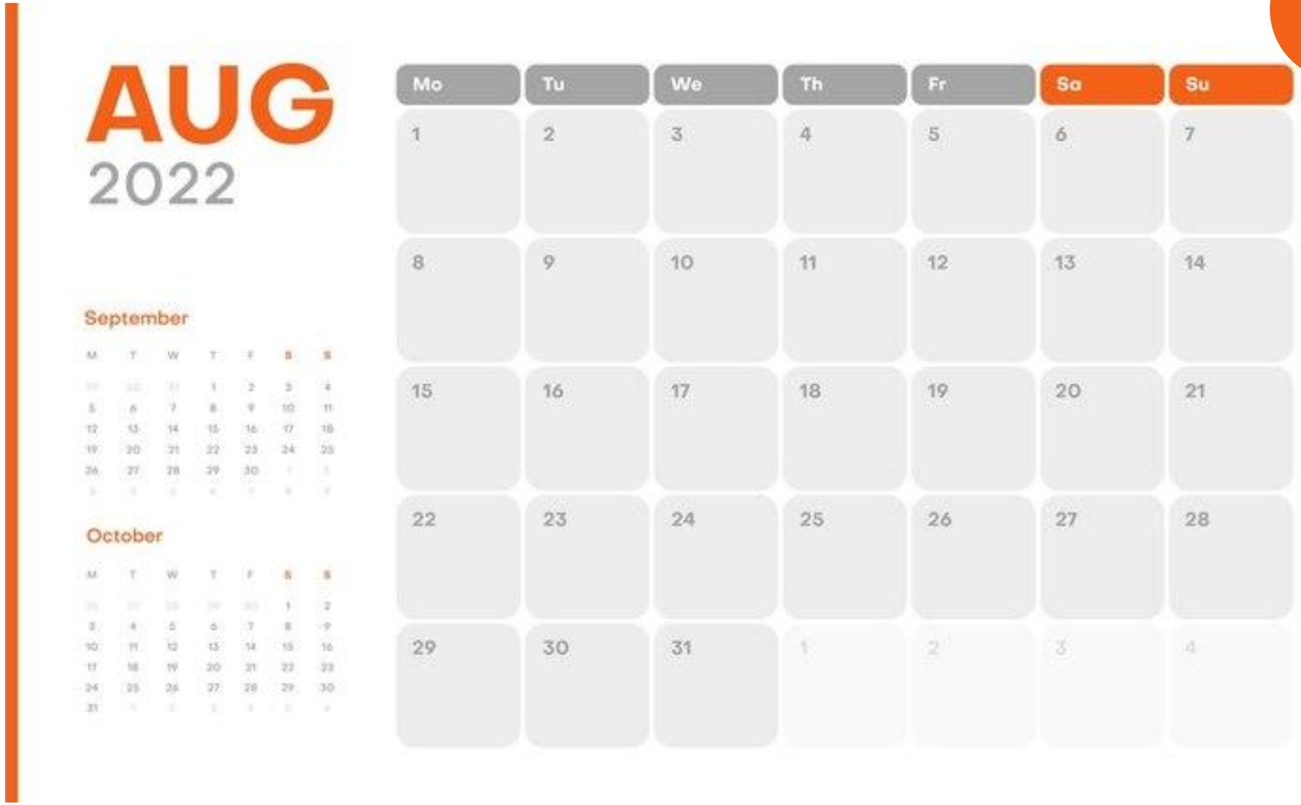
## App v2: FizzBuzz Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Otherwise, return None.

```python
def fun_notification(n):
    response = ''
    if not n % 3:
        response += 'Fizz'
    if not n % 5:
        response += 'Buzz'
    return response if response else None
```

## App v2: FizzBuzz Feature

2



AUG
2022

| Mo | Tu | We | Th | Fr | Sa | Su |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |

September

| M | T | W | T | F | S | S |
|---|---|---|---|---|---|---|
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |

October

| M | T | W | T | F | S | S |
|---|---|---|---|---|---|---|
| 26 | 27 | 28 | 29 | 30 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 1 | 2 | 3 | 4 | 5 | 6 |

## App v2: Calendar App API

Full Fun Notification Feature

Bencher

## App v3: FizzBuzzFibonacci Feature

## App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both.

## App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence.

## App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return n$^{th}$ step of the Fibonacci Sequence. Otherwise, return None.

# App v3: FizzBuzzFibonacci Feature

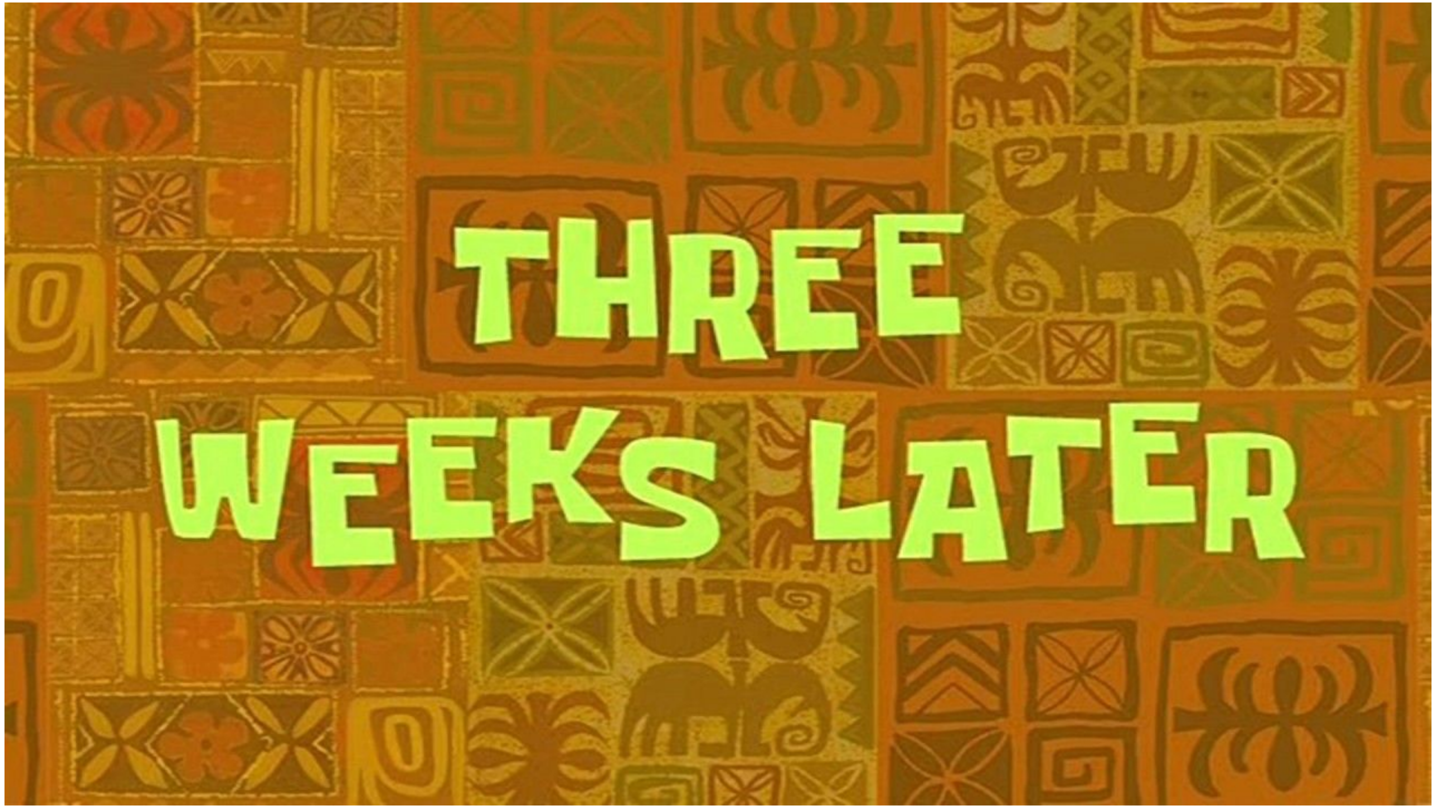
Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence. Otherwise, return None.

```python
def fun_notification(n):
    if not n % 7:
        return fibonacci(n)
    response = ''
    if not n % 3:
        response += 'Fizz'
    if not n % 5:
        response += 'Buzz'
    return response if response else None
```

Bencher

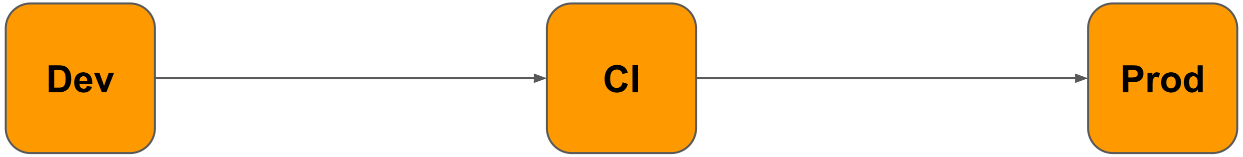
# App v3: FizzBuzzFibonacci Feature

3

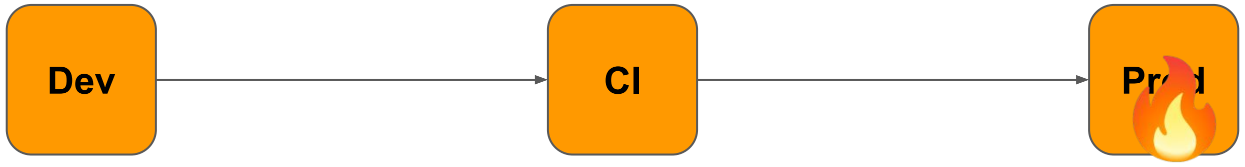


Bencher

## When do performance regression get detected?



## When do performance regression get detected?



# App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence. Otherwise, return None.

```python
def fun_notification(n):
    if not n % 7:
        return fibonacci(n)
    response = ''
    if not n % 3:
        response += 'Fizz'
    if not n % 5:
        response += 'Buzz'
    return response if response else None
```

## App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence. Otherwise, return None.

```python
def fun_notification(n):
    if not n % 7:
        return fibonacci(n)
    response = ''
    if not n % 3:
        response += 'Fizz'
    if not n % 5:
        response += 'Buzz'
    return response if response else None
```

Bencher

# App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence. Otherwise, return None.

```python
def fun_notification(n):
    if not n % 7:
        return fibonacci(n)
    response = ''
    if not n % 3:
        response += 'Fizz'
    if not n % 5:
        response += 'Buzz'
    return response if response else None
```

Bencher

## App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence. Otherwise, return None.

```python
def fibonacci(n):
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)
```

Bencher

## Benchmarking in Python

Bencher

## Benchmarking in Python

pytest-benchmark

## Benchmarking in Python

pytest-benchmark

airspeed velocity (asv)

## Install pytest-benchmark

```
pipenv shell
pip install pytest-benchmark
```

Bencher

## fun_notifcation.py

```python
def fibonacci(n):
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)
```

fun_notifcation.py

```python
def fibonacci(n):
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)


def test_fibonacci(benchmark):
    def fibonacci_month():
        for n in range(7, 29, 7):
            fibonacci(n)
    benchmark(fibonacci_month)
```

Bencher

## fun_notifcation.py

```python
def fibonacci(n):
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)


def test_fibonacci(benchmark):
    def fibonacci_month():
        for n in range(7, 29, 7):
            fibonacci(n)
    benchmark(fibonacci_month)
```

Bencher

## fun_notifcation.py

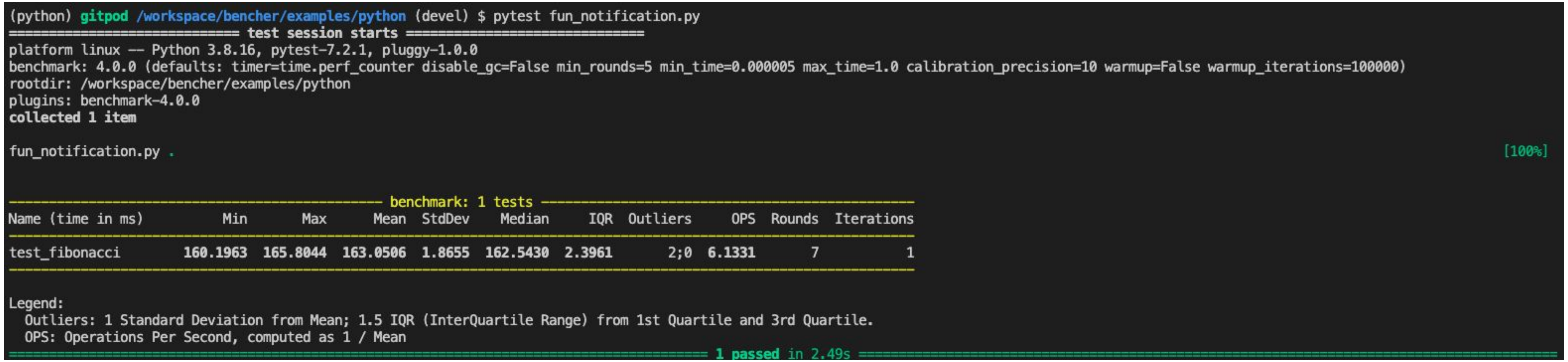
```python
def fibonacci(n):
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)


def test_fibonacci(benchmark):
    def fibonacci_month():
        for n in range(7, 29, 7):
            fibonacci(n)
    benchmark(fibonacci_month)
```

Bencher

## Run pytest-benchmark

```
pytest fun_notification.py
```

# Run pytest-benchmark



```
(python) gitpod /workspace/bencher/examples/python (devel) $ pytest fun_notification.py
============================= test session starts ==============================
platform linux -- Python 3.8.16, pytest-7.2.1, pluggy-1.0.0
benchmark: 4.0.0 (defaults: timer=time.perf_counter disable_gc=False min_rounds=5 min_time=0.000005 max_time=1.0 calibration_precision=10 warmup=False warmup_iterations=100000)
rootdir: /workspace/bencher/examples/python
plugins: benchmark-4.0.0
collected 1 item

fun_notification.py .                                                                                                                                                                    [100%]


----------------------------------------------- benchmark: 1 tests -----------------------------------------------
Name (time in ms)         Min       Max      Mean  StdDev    Median     IQR  Outliers     OPS  Rounds  Iterations
-----------------------------------------------------------------------------------------------------------------
test_fibonacci       160.1963  165.8044  163.0506  1.8655  162.5430  2.3961       2;0  6.1331       7           1
-----------------------------------------------------------------------------------------------------------------

Legend:
  Outliers: 1 Standard Deviation from Mean; 1.5 IQR (InterQuartile Range) from 1st Quartile and 3rd Quartile.
  OPS: Operations Per Second, computed as 1 / Mean
======================================================================================== 1 passed in 2.49s ========================================================================================
```

Bencher

## Run and Save pytest-benchmark

```
pytest --benchmark-autosave fun_notification.py
```

Bencher

## fun_notifcation.py

```
def fibonacci(n):
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)


def test_fibonacci(benchmark):
    def fibonacci_month():
        for n in range(7, 29, 7):
            fibonacci(n)
    benchmark(fibonacci_month)
```

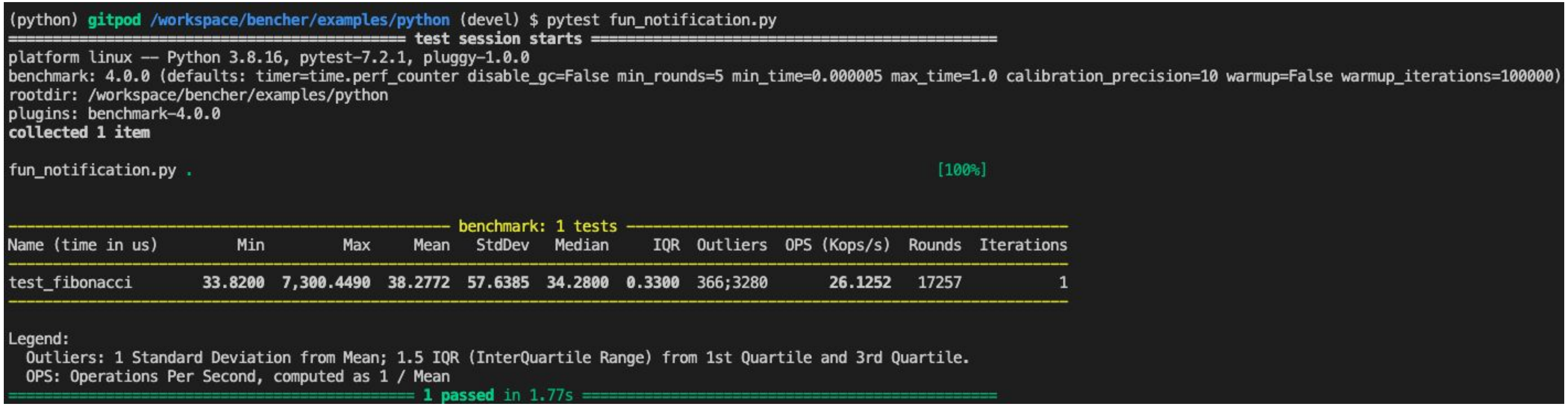
## fun_notifcation.py

```python
def fibonacci(n):
    pad = {0: 0, 1: 1}
    def memo(n):
        if n not in pad:
            pad[n] = memo(n-1) + memo(n-2)
        return pad[n]
    return memo(n)


def test_fibonacci(benchmark):
    def fibonacci_month():
        for n in range(7, 29, 7):
            fibonacci(n)
    benchmark(fibonacci_month)
```

## Run pytest-benchmark

```
pytest fun_notification.py
```

# Run pytest-benchmark

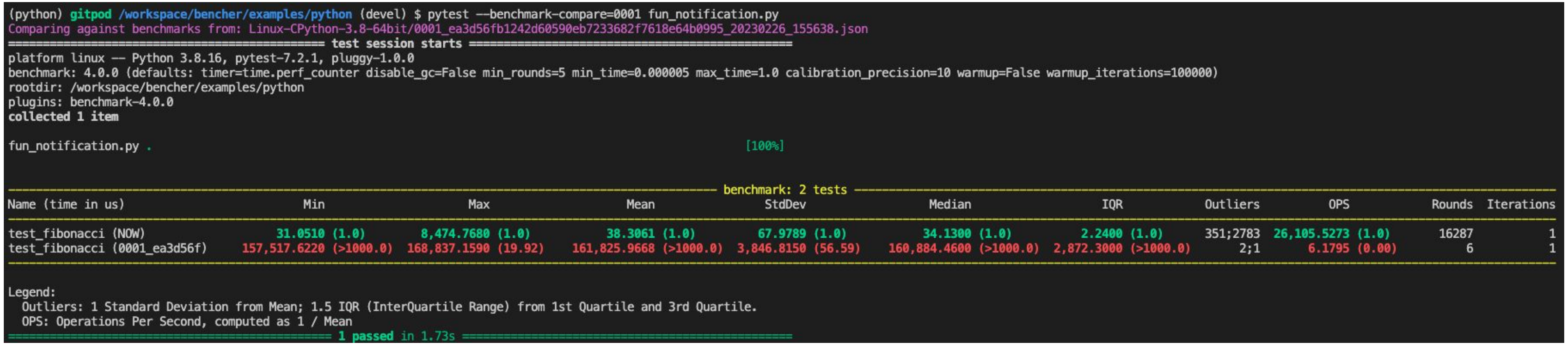


```
(python) gitpod /workspace/bencher/examples/python (devel) $ pytest fun_notification.py
============================================== test session starts ==============================================
platform linux -- Python 3.8.16, pytest-7.2.1, pluggy-1.0.0
benchmark: 4.0.0 (defaults: timer=time.perf_counter disable_gc=False min_rounds=5 min_time=0.000005 max_time=1.0 calibration_precision=10 warmup=False warmup_iterations=100000)
rootdir: /workspace/bencher/examples/python
plugins: benchmark-4.0.0
collected 1 item

fun_notification.py .                                                                                     [100%]


-------------------------------------------------- benchmark: 1 tests --------------------------------------------------
Name (time in us)        Min          Max     Mean   StdDev   Median     IQR  Outliers  OPS (Kops/s)  Rounds  Iterations
-------------------------------------------------------------------------------------------------------------------------
test_fibonacci       33.8200  7,300.4490  38.2772  57.6385  34.2800  0.3300  366;3280       26.1252   17257           1
-------------------------------------------------------------------------------------------------------------------------


Legend:
  Outliers: 1 Standard Deviation from Mean; 1.5 IQR (InterQuartile Range) from 1st Quartile and 3rd Quartile.
  OPS: Operations Per Second, computed as 1 / Mean
============================================== 1 passed in 1.77s ===============================================
```

## Run and Compare with pytest-benchmark

```
pytest --benchmark-compare=0001 fun_notification.py
```

# Run and Compare with pytest-benchmark



```
(python) gitpod /workspace/bencher/examples/python (devel) $ pytest --benchmark-compare=0001 fun_notification.py
Comparing against benchmarks from: Linux-CPython-3.8-64bit/0001_ea3d56fb1242d60590eb7233682f7618e64b0995_20230226_155638.json
============================================== test session starts ==============================================
platform linux -- Python 3.8.16, pytest-7.2.1, pluggy-1.0.0
benchmark: 4.0.0 (defaults: timer=time.perf_counter disable_gc=False min_rounds=5 min_time=0.000005 max_time=1.0 calibration_precision=10 warmup=False warmup_iterations=100000)
rootdir: /workspace/bencher/examples/python
plugins: benchmark-4.0.0
collected 1 item

fun_notification.py .                                                                                     [100%]


------------------------------------------------------------------------------------------------------- benchmark: 2 tests -------------------------------------------------------------------------------------------------------
Name (time in us)                        Min                      Max                        Mean                 StdDev                      Median                     IQR            Outliers         OPS            Rounds  Iterations
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
test_fibonacci (NOW)                 31.0510 (1.0)        8,474.7680 (1.0)          38.3061 (1.0)         67.9789 (1.0)            34.1300 (1.0)            2.2400 (1.0)       351;2783  26,105.5273 (1.0)      16287           1
test_fibonacci (0001_ea3d56f)   157,517.6220 (>1000.0)  168,837.1590 (19.92)   161,825.9668 (>1000.0)  3,846.8150 (56.59)   160,884.4600 (>1000.0)  2,872.3000 (>1000.0)       2;1       6.1795 (0.00)          6           1
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Legend:
  Outliers: 1 Standard Deviation from Mean; 1.5 IQR (InterQuartile Range) from 1st Quartile and 3rd Quartile.
  OPS: Operations Per Second, computed as 1 / Mean
=============================================== 1 passed in 1.73s ===============================================
```

Bencher

## Micro vs Macro Benchmarks

## Micro vs Macro Benchmarks

Micro
(unit)

```
def fibonacci(n):
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)
```

## Micro vs Macro Benchmarks

Micro
(unit)

```
def fibonacci(n):
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)
```

Macro
(integration)

```
@ping_blueprint_v1.route("/api/v1/fun", methods=["GET"])
def fun_notification():
    day_of_month = datetime.now().day
    notification = fun_notification(day_of_month)
    return jsonify({"status": "success", "message": notification})
```

Bencher

## App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence. Otherwise, return None.

```python
def fibonacci(n):
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)
```

Bencher

## App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence. Otherwise, return None.

```
def fibonacci(n):
    if n < 2:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)
```

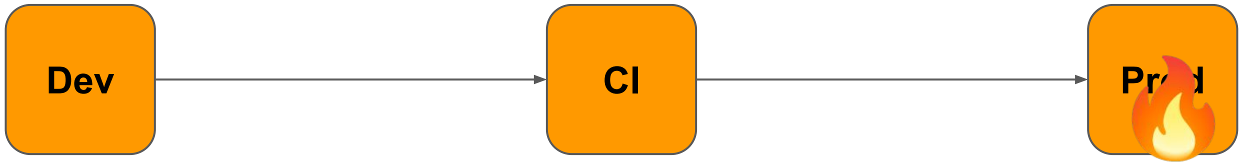
## App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence. Otherwise, return None.

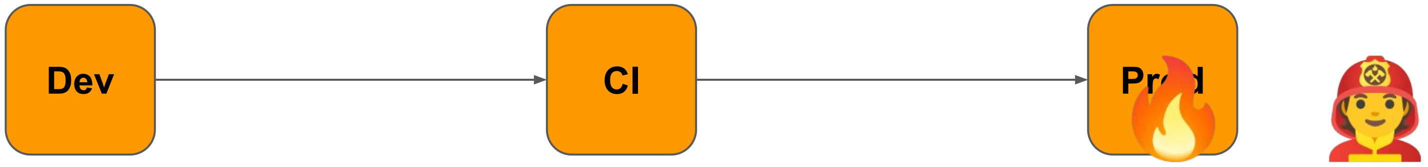
```
def fibonacci(n):
    pad = {0: 0, 1: 1}
    def memo(n):
        if n not in pad:
            pad[n] = memo(n-1) + memo(n-2)
        return pad[n]
    return memo(n)
```

Bencher

## When do performance regression get detected?

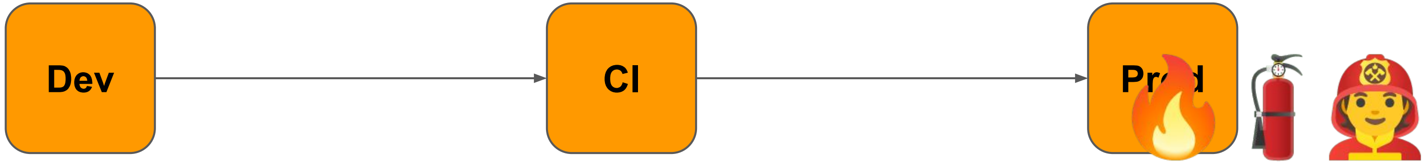


## When do performance regression get detected?

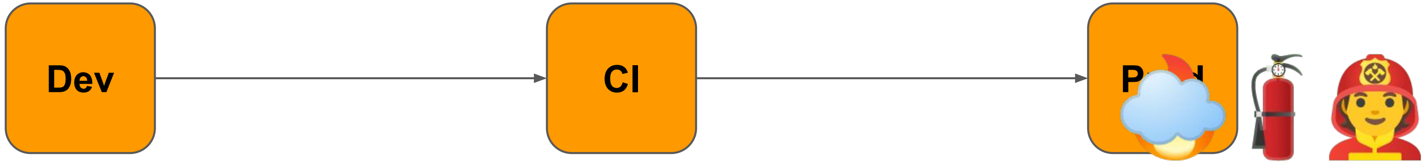


Bencher

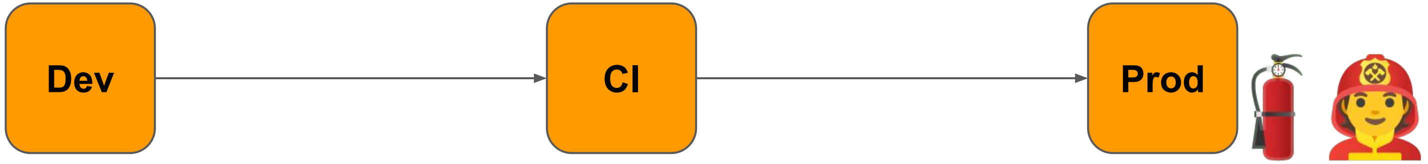
## When do performance regression get detected?



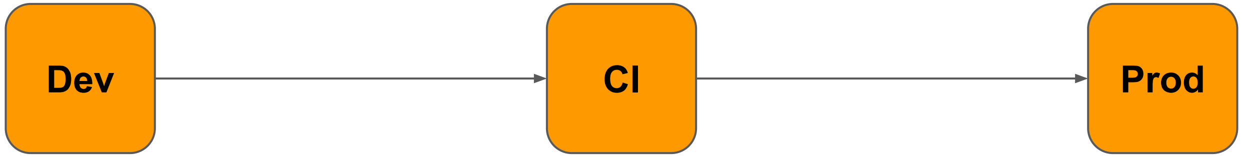
## When do performance regression get detected?



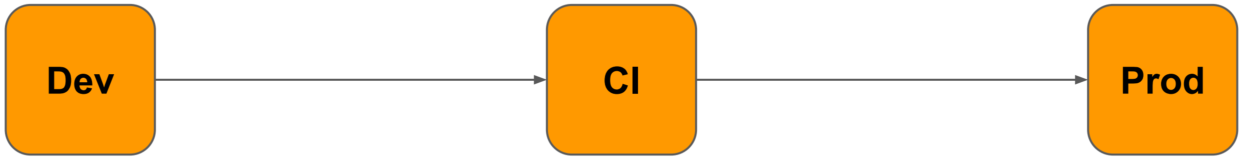
## When do performance regression get detected?



## When do performance regression get detected?



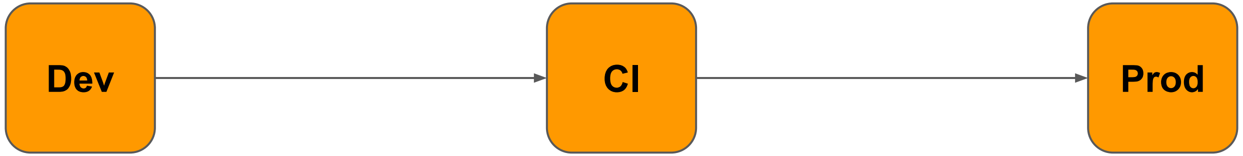
## When do performance regression get detected?



Observability Tools?

❌ Too Late

## When do performance regression get detected?



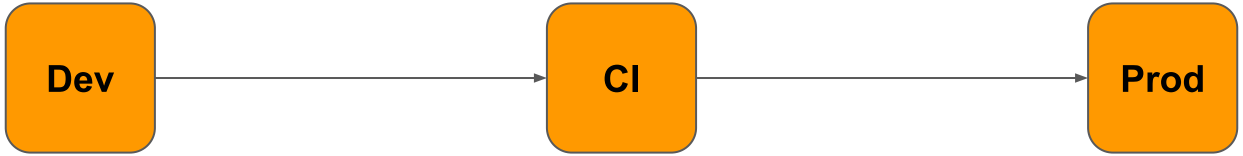
Local benchmark comparison
- pytest-benchmark
- airspeed velocity (asv)

❌ Local Only

Observability Tools?

❌ Too Late

## When do performance regression get detected?



Local benchmark comparison
- pytest-benchmark
- airspeed velocity (asv)

❌ Local Only

Continuous Benchmarking
- Bencher
- airspeed velocity (asv)

Observability Tools?

❌ Too Late

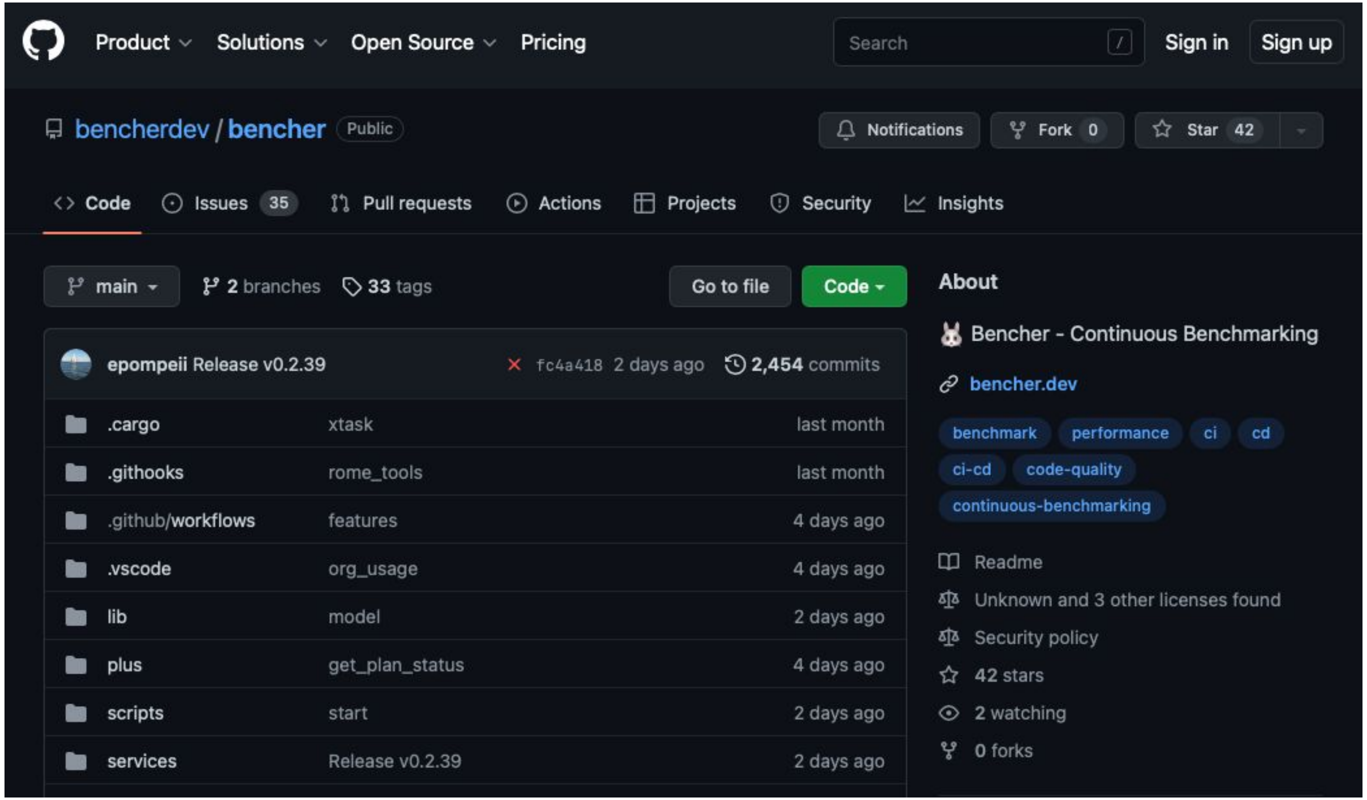
## What if you had Continuous Benchmarking?

Bencher

## What if you had Continuous Benchmarking?

Bencher

# What if you had Continuous Benchmarking?



bencherdev / bencher Public

2 branches 33 tags

epompeii Release v0.2.39 fc4a418 2 days ago 2,454 commits

| | | |
|---|---|---|
| .cargo | xtask | last month |
| .githooks | rome_tools | last month |
| .github/workflows | features | 4 days ago |
| .vscode | org_usage | 4 days ago |
| lib | model | 2 days ago |
| plus | get_plan_status | 4 days ago |
| scripts | start | 2 days ago |
| services | Release v0.2.39 | 2 days ago |

About

Bencher - Continuous Benchmarking

bencher.dev

benchmark performance ci cd ci-cd code-quality continuous-benchmarking

Readme

Unknown and 3 other licenses found

Security policy

42 stars

2 watching

0 forks

Bencher

## App v1: Fizz Feature

Return “Fizz” if day is divisible by 3.
Otherwise, return None.

```python
def fun_notification(n):
    if not n % 3:
        return 'Fizz'
    return None
```

Bencher

## App v1: Fizz Feature

Return “Fizz” if day is divisible by 3.
Otherwise, return None.

```python
def fun_notification(n):
    if not n % 3:
        return 'Fizz'
    return None
```

```python
def test_fun_notification(benchmark):
    def days_in_month():
        for n in range(1, 32):
            fun_notification(n)
    benchmark(days_in_month)
```

Bencher

## Install Bencher CLI in CI

```
wget https://github.com/bencherdev/bencher/releases/download/v0.2.40/bencher_0.2.40_amd64.deb
sudo dpkg -i bencher_0.2.40_amd64.deb
```

Bencher

## Continuous Benchmarking with pytest-benchmark

```
bencher run \
    --file results.json \
    “pipenv run pytest \
        --benchmark-json results.json \
        fun_notification.py”
```

## App v1: Fizz Feature

Return “Fizz” if day is divisible by 3.
Otherwise, return None.

```python
def fun_notification(n):
    if not n % 3:
        return 'Fizz'
    return None
```

```python
def test_fun_notification(benchmark):
    def days_in_month():
        for n in range(1, 32):
            fun_notification(n)
    benchmark(days_in_month)
```

## App v2: FizzBuzz Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Otherwise, return None.

```python
def fun_notification(n):
    response = ''
    if not n % 3:
        response += 'Fizz'
    if not n % 5:
        response += 'Buzz'
    return response if response else None
```

```python
def test_fun_notification(benchmark):
    def days_in_month():
        for n in range(1, 32):
            fun_notification(n)
    benchmark(days_in_month)
```

Bencher

## App v3: FizzBuzzFibonacci Feature

Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence. Otherwise, return None.

```python
def fun_notification(n):
    if not n % 7:
        return fibonacci(n)
    response = ''
    if not n % 3:
        response += 'Fizz'
    if not n % 5:
        response += 'Buzz'
    return response if response else None
```

```python
def test_fun_notification(benchmark):
    def days_in_month():
        for n in range(1, 32):
            fun_notification(n)
    benchmark(days_in_month)
```

Bencher

# App v3: FizzBuzzFibonacci Feature

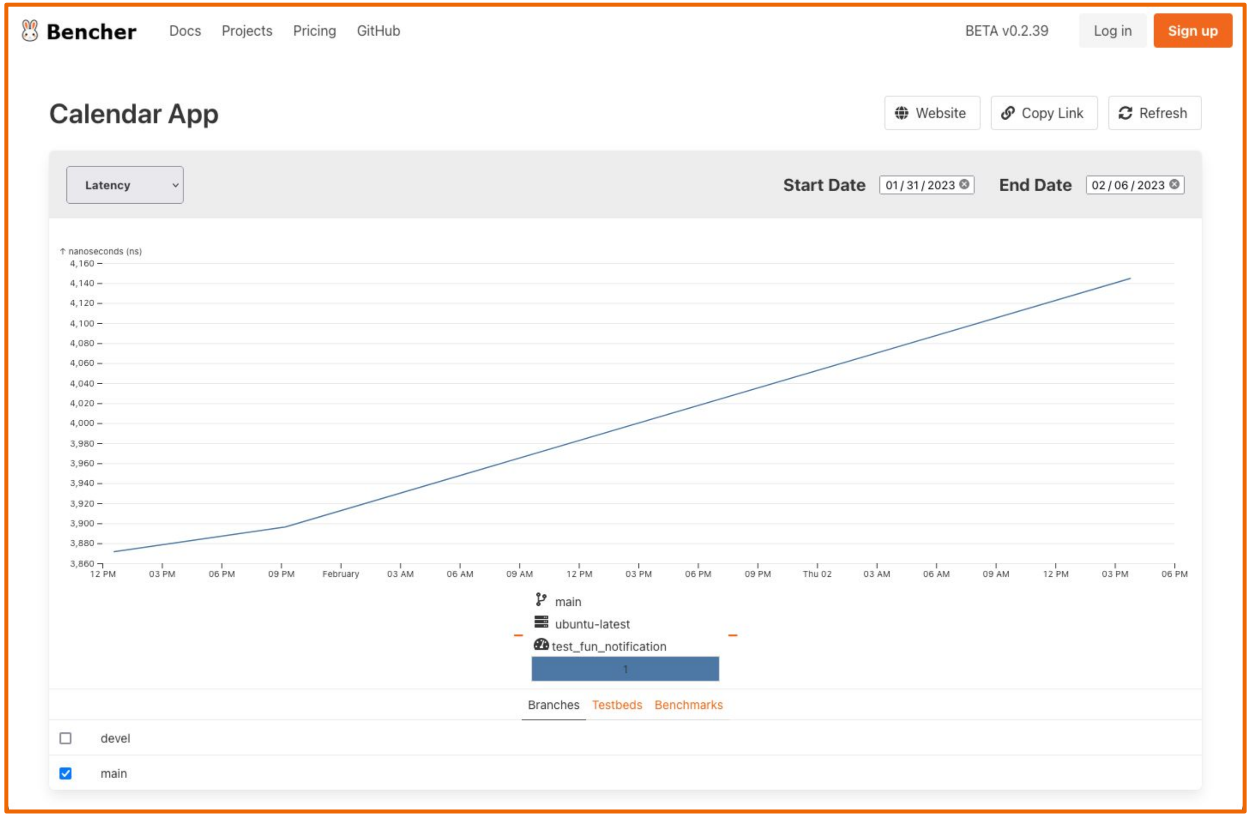
Return “Fizz” if day is divisible by 3, “Buzz” if divisible by 5, or “FizzBuzz” if both. Except if day is divisible by 7, then return nth step of the Fibonacci Sequence. Otherwise, return None.

```python
def fun_notification(n):
    if not n % 7:
        return fibonacci(n)
    response = ''
    if not n % 3:
        response += 'Fizz'
    if not n % 5:
        response += 'Buzz'
    return response if response else None
```

```python
def test_fun_notification(benchmark):
    def days_in_month():
        for n in range(1, 32):
            fun_notification(n)
    benchmark(days_in_month)
```

🚨

## Track Your Benchmarks



Calendar App

Start Date 01 / 31 / 2023

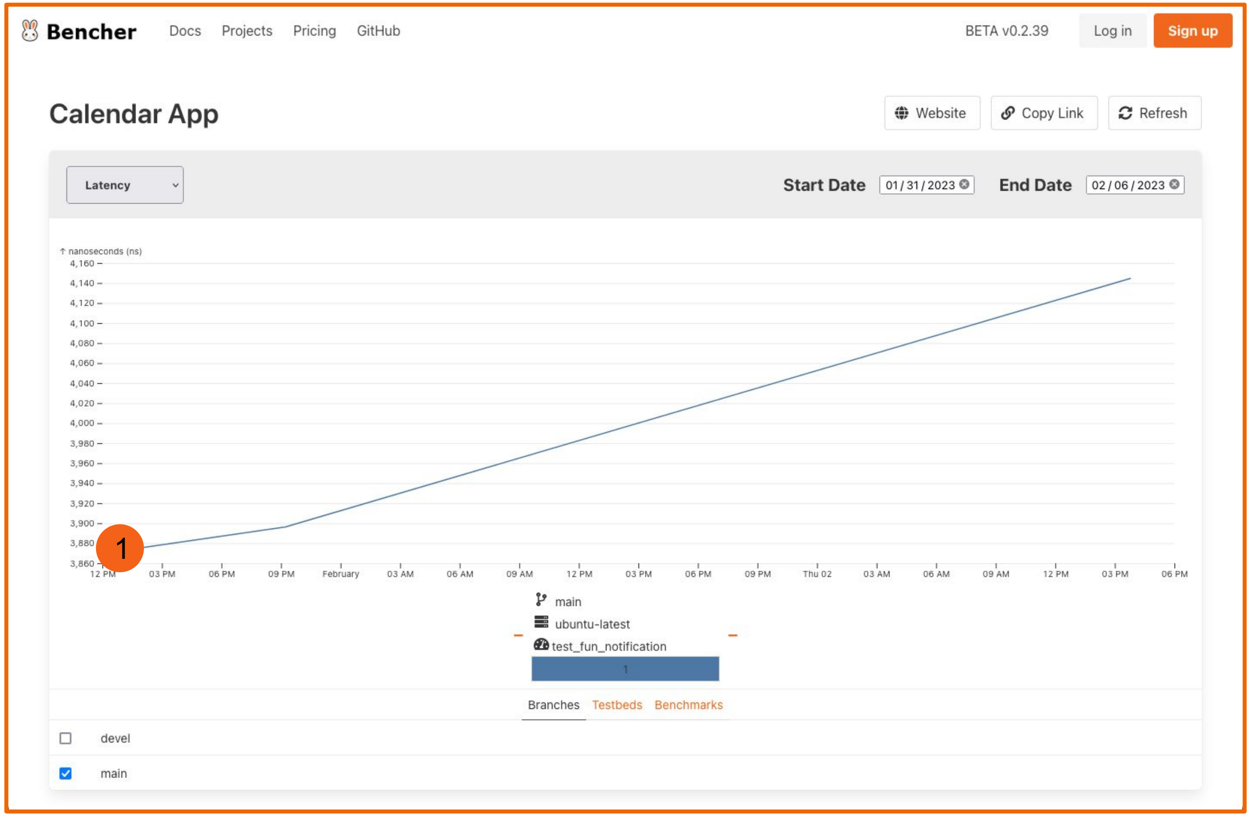
End Date 02 / 06 / 2023

☐ devel

☑ main

Bencher

## Track Your Benchmarks



Calendar App

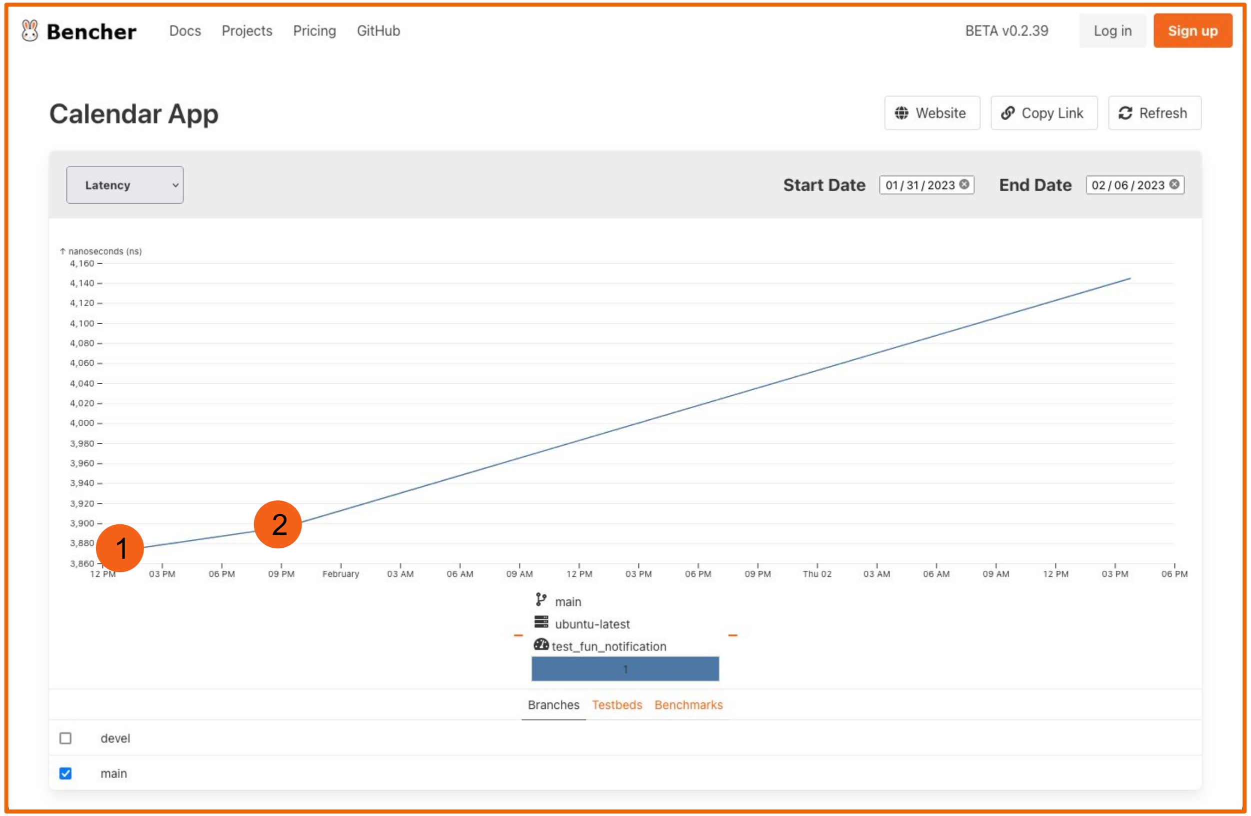
Start Date 01 / 31 / 2023 End Date 02 / 06 / 2023

☐ devel

☑ main

Bencher

## Track Your Benchmarks



Bencher Docs Projects Pricing GitHub BETA v0.2.39 Log in Sign up

Calendar App

Website Copy Link Refresh

Latency

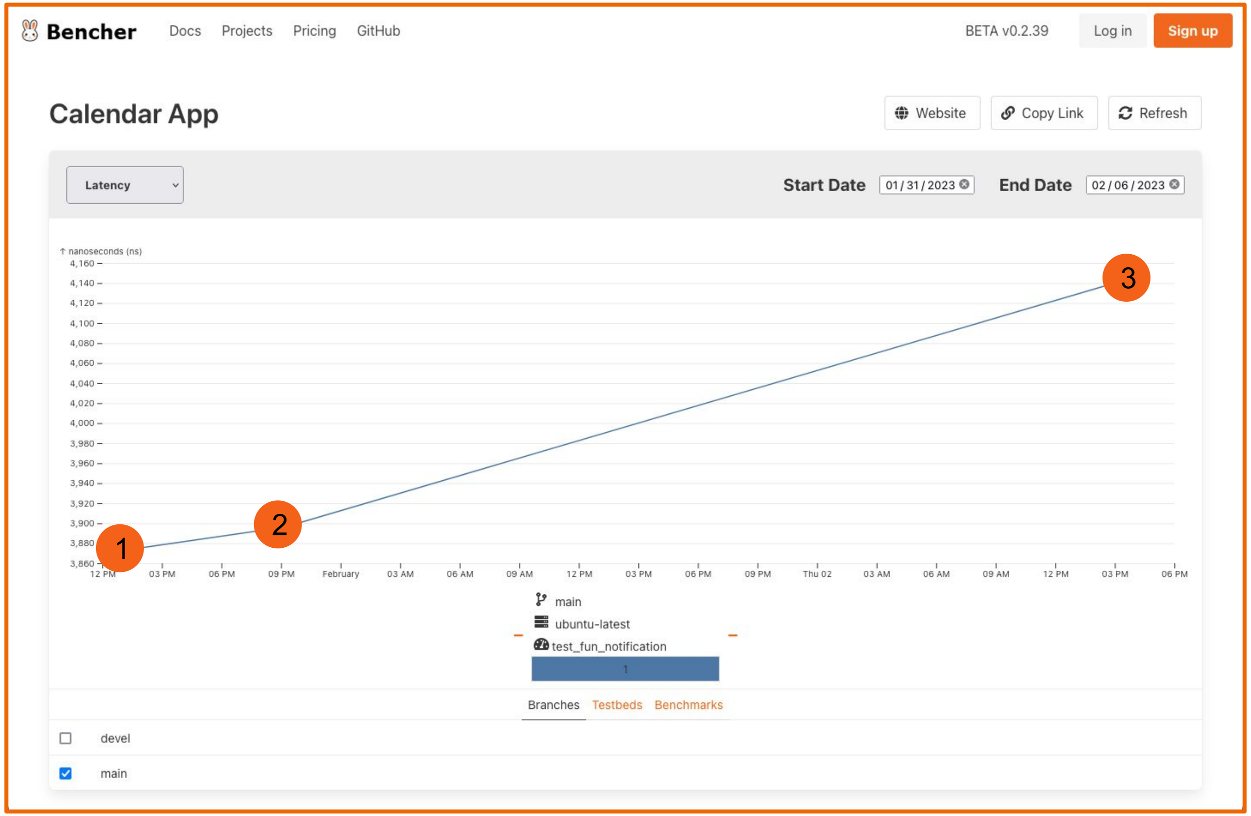
Start Date 01/31/2023 End Date 02/06/2023

Branches Testbeds Benchmarks

- [ ] devel
- [x] main

Bencher

## Track Your Benchmarks



Calendar App

Latency

Start Date 01 / 31 / 2023

End Date 02 / 06 / 2023

main

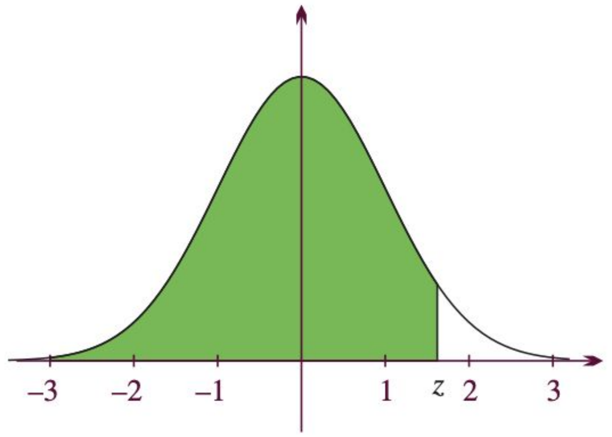
ubuntu-latest

test_fun_notification

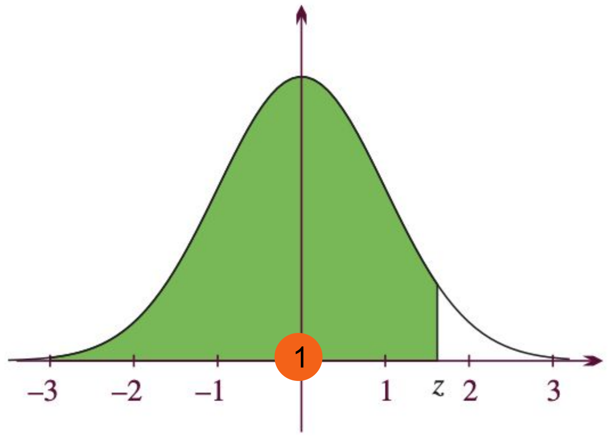
Branches Testbeds Benchmarks

| | |
|---|---|
| ☐ | devel |
| ☑ | main |

## Performance Regression Alerts

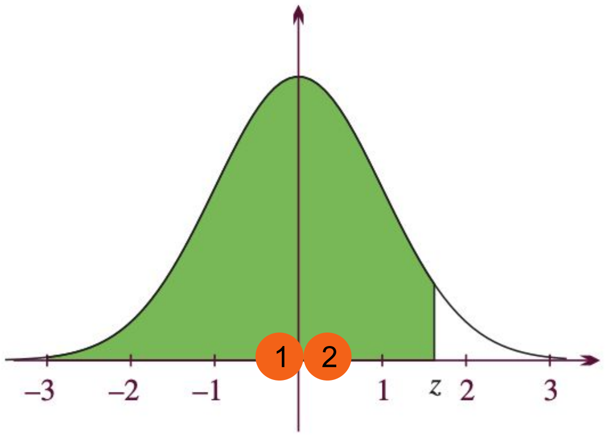


# Performance Regression Alerts

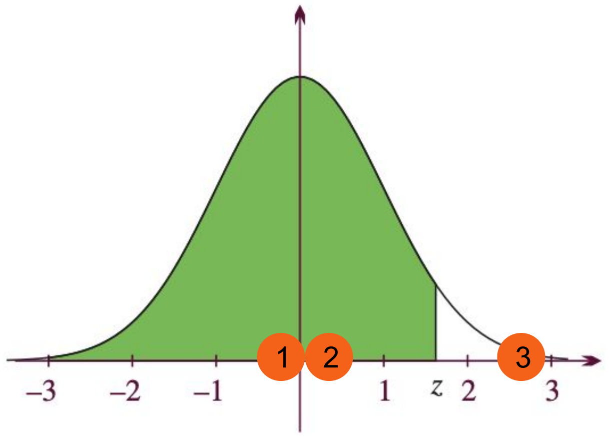


Bencher

# Performance Regression Alerts

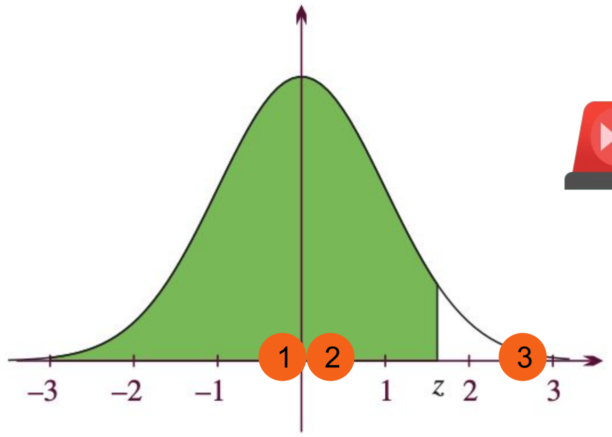


## Performance Regression Alerts



Bencher

## Performance Regression Alerts



## Catch performance regressions in CI

Bencher

## When do performance regression get detected?

$$\begin{tikzcd}\boxed{\textbf{Dev}} \arrow[r] & \boxed{\textbf{CI}} \arrow[r] & \boxed{\textbf{Prod}}\end{tikzcd}$$

Local benchmark comparison

- pytest-benchmark
- airspeed velocity (asv)

❌ Local Only

Continuous Benchmarking

- Bencher
- airspeed velocity (asv)

Observability Tools?

❌ Too Late

## When do performance regression get detected?

$$\begin{tikzcd}\textbf{Dev} \arrow[r] & \textbf{CI} \arrow[r] & \textbf{Prod}\end{tikzcd}$$

Local benchmark comparison

- pytest-benchmark
- airspeed velocity (asv)

❌ Local Only

Continuous Benchmarking

- Bencher
- airspeed velocity (asv)

🐰 Awesome!

Observability Tools?

❌ Too Late

## Review

Bencher

## Review

- Detection ➡ Prevention

## Review

- Detection ➡ Prevention
- Production is too late

## Review

- Detection ➡ Prevention
- Production is too late
- Development is local only

## Review

- Detection ➡ Prevention
- Production is too late
- Development is local only
- Continuous Benchmarking can save us a lot of pain

## 🐍 Run Fast!

Catch Performance Regressions in Python

https://github.com/bencherdev/bencher

Bencher

## 🐍 Run Fast!

Catch Performance Regressions in Python

https://github.com/bencherdev/bencher

# 🐍 Run Fast!

Catch Performance Regressions in Python

https://github.com/bencherdev/bencher

bencher.dev/repo

Bencher