



PiterPy

Инфраструктурный Python

Николай Марков, 2023

Shameless Plug

- Меня зовут [Николай Марков](#), и мой ник [@enchantner](#) практически везде в интернете
- Я Principal Architect в компании [Rayo Data](#) / DE Platform Lead в [Altenar](#)
- Больше 13 лет пишу на Python, а также использую Golang, C/C++, Rust и Scala
- Ковыряюсь в облаках, распределенке и сетях (AWS, Kubernetes)
- Преподаю в [разных компаниях](#) и [странах](#). Пишу [статьи](#), участвую в программных [комитетах мероприятий](#)
- Люблю болтать о системном дизайне, архитектуре и прочей низкоуровневой ерунде



Два базовых вопроса в чате



Как в моем проекте решить вот эту штуку, которую уже все сто раз делали, но на Python?

Куда бы мне поконтрибьютить на Python, чтобы получить опыт?

Язык общего назначения

Python ни разу не мертв и до сих пор решает массу задач и за пределами Data Science

У нас лучше, чем в Javascript - стабильные большие проекты, зрелая экосистема, низкий уровень

У всех есть время собирать ссылки, теперь настало время разбрасывать ссылки



Я собираю ссылки

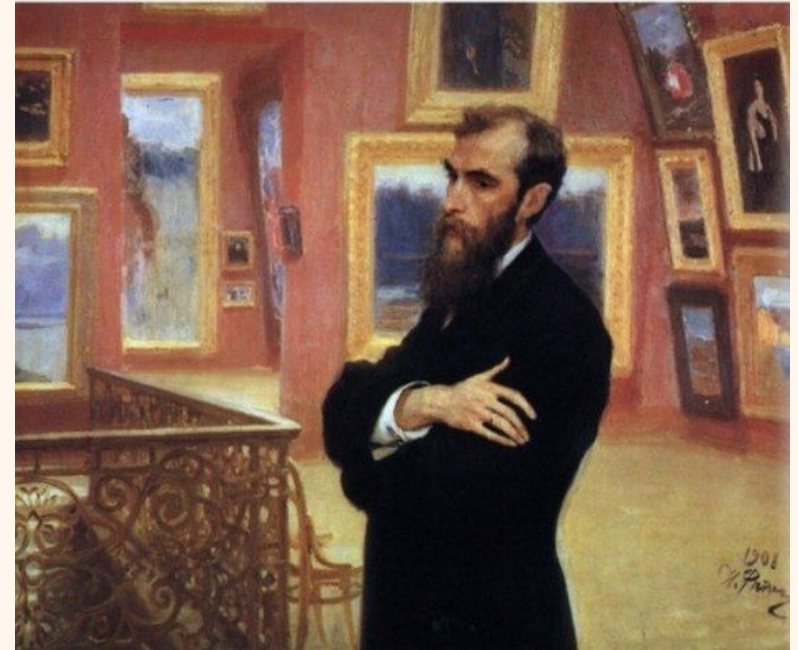
В СВЯЗИ С ДЕФОЛТОМ ДОЛЛАРА ПРАВИТЕЛЬСТВО
ВВЕЛО АЛЬТЕРНАТИВНУЮ ВАЛЮТУ.

ВАШЕ МАТЕРИАЛЬНОЕ БЛАГОСОСТОЯНИЕ ТЕПЕРЬ
ОПРЕДЕЛЯЕТСЯ КОЛИЧЕСТВОМ СМЕШНЫХ КАРТИНОК,
СОХРАНЁННЫХ НА ВАШЕМ ЖЁСТКОМ ДИСКЕ.



Я ГОТОВИЛСЯ К ЭТОМУ МОМЕНТУ ВСЮ СВОЮ ЖИЗНЬ.

Не расстраивайся, Павел Михайлович Третьяков
занимался тем же, что и ты - сохранял
понравившиеся картинки, показывал друзьям -
и теперь, спустя много лет, все его знают!



Пишем новый проект

Мы на необитаемом острове, где есть
только питоны

Нужно написать хайповое
приложение, чтобы нам дали денег и
спасли

Запустим соцсеть для любителей
соево-кокосового рафа

Все используемые проекты должны быть
живыми и иметь звезды на гитхабе!

+ все логотипы и ссылки – кликабельные



Как выжить с одним питоном

Как выстроить процесс разработки?

Как избежать бойлерплейта?

Как организовать CI/CD?

Как запускать периодические и отложенные задания?

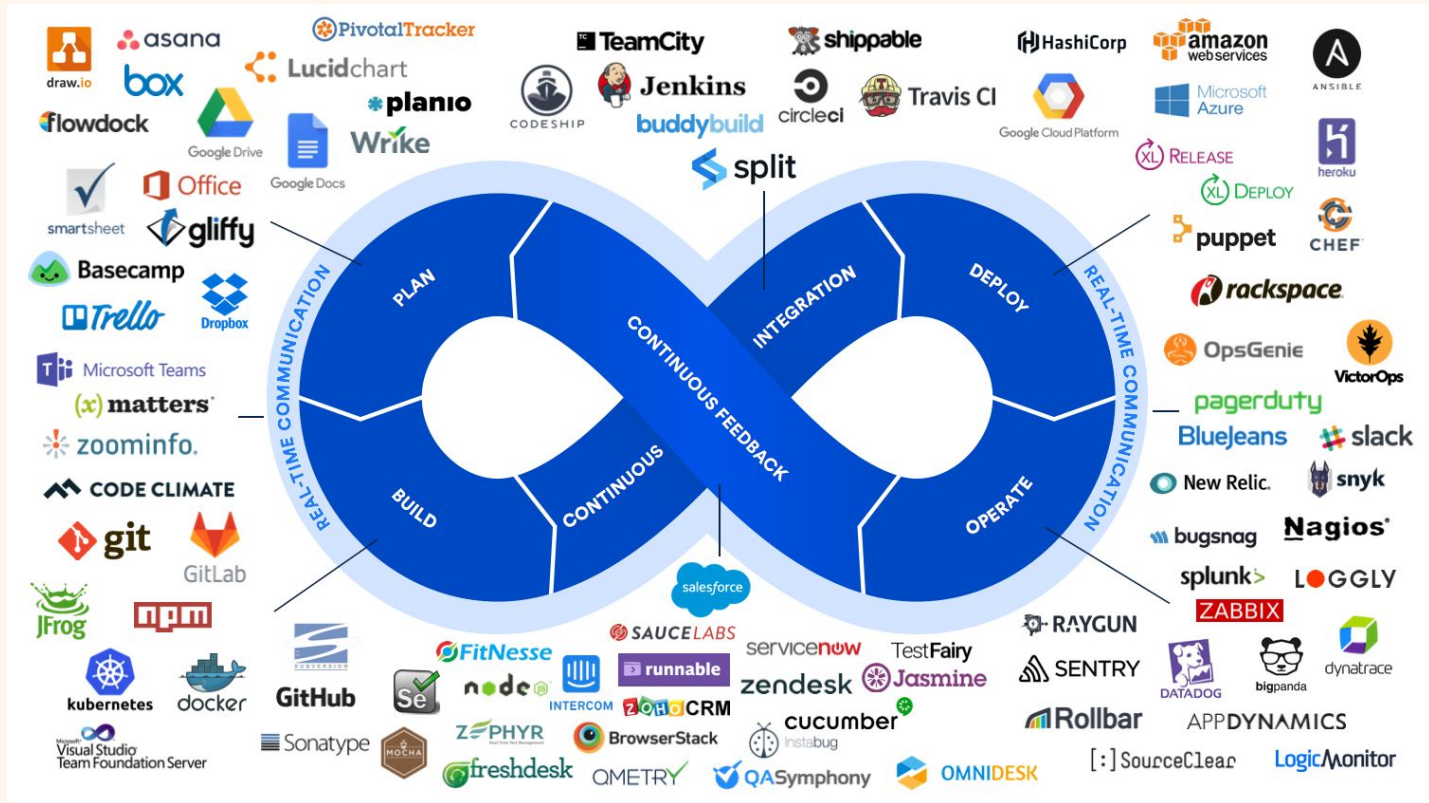
Как мониторить метрики?

Как тестировать?

Как использовать облака?

Как управлять инфраструктурой?

DevOps



Система контроля версий



git



Система контроля версий



git

★ 47.9k



★ 533

Система контроля версий



git

★ 47.9k

используется в
Facebook и Mozilla
([Project Sapling](#))



mercurial

★ ?



★ 533

[Рефлексия по поводу
перехода на Python 3 и
Rust](#)

Уголок здравого смысла

используйте



git

Pre-commit хуки

<https://github.com/pre-commit/pre-commit>

```
1. repos:
2. - repo: https://github.com/pre-commit/pre-commit-hooks
3.   rev: v4.5.0
4.   hooks:
5.     - id: trailing-whitespace
6.     - id: end-of-file-fixer
7.     - id: check-yaml
8.     - id: debug-statements
9.     - id: double-quote-string-fixer
10.    - id: name-tests-test
11.    - id: requirements-txt-fixer
12. - repo: https://github.com/asottile/setup-cfg-fmt
13.   rev: v2.5.0
14.   hooks:
15.     - id: setup-cfg-fmt
16. - repo: https://github.com/asottile/reorder-python-imports
17.   rev: v3.12.0
18.   hooks:
19.     - id: reorder-python-imports
20.   exclude: ^(pre_commit/resources/|testing/resources/python3_hooks_repo/)
21.   args: [--py38-plus, --add-import, 'from __future__ import annotations']
```



Cookiecutter + Scaraplate

```
{
  "full_name": "Audrey Roy Greenfeld",
  "email": "aroy@alum.mit.edu",
  "github_username": "audreyr",
  "project_name": "Python Boilerplate",
  "project_slug": "{{ cookiecutter.project_name.lower().replace(' ', '_') }}",
  "project_short_description": "Python Boilerplate contains all the boilerplate you
need to create a Python package.",
  "pypi_username": "{{ cookiecutter.github_username }}",
  "version": "0.1.0",
  "use_pytest": "n",
  "use_pypi_deployment_with_travis": "y",
  "create_author_file": "y",
  "open_source_license": ["MIT", "BSD", "ISCL", "Apache Software License 2.0",
"Not open source"]
}
```

скелет проекта нашей соцсети

Конфигурирование приложения


<https://12factor.net/ru/>



Аргументы?
Переменные окружения?
Формат конфига?



 **TypeORM**
★ 12.4k

\$ click_  ★ 14.5k

запуск сервера соцсети

Конфигурирование приложения 2.0



```
db:  
  driver: mysql  
  pass: secret  
  user: omry
```

```
$ python my_app.py db.user=root db.pass=1234  
db:  
  driver: mysql  
  user: root  
  pass: 1234
```

```
1. import hydra  
2. from omegaconf import DictConfig, OmegaConf  
3.  
4. @hydra.main(version_base=None, config_path="conf", config_name="config")  
5. def my_app(cfg : DictConfig) -> None:  
6.     print(OmegaConf.to_yaml(cfg))  
7.  
8. if __name__ == "__main__":  
9.     my_app()
```


Конфигурирование приложения 2.0



★ 3.3k

config.py

```
from dynaconf import Dynaconf

settings = Dynaconf(
    settings_files=[
        'settings.toml',
        '.secrets.toml'
    ],
)
```

```
export DYNACONF_NUMBER=789
export DYNACONF_FOO=false
export DYNACONF_DATA_CAN_BE_NESTED=value
export DYNACONF_FORMATTED_KEY="@format {this.FOO}/BAR"
export DYNACONF_TEMPLATED_KEY="@jinja {{ env['HOME'] | abspath }}"
```

code.py

```
from config import settings

assert settings.key == "value"
assert settings.number == 789
assert settings.a_dict.nested.other_level == "nested value"
assert settings['a_boolean'] is False
assert settings.get("DONTEXIST", default=1) == 1
```

Единообразие конфигов везде - [Nitpick](#)

★ 366

```
["pyproject.toml".tool.black]
line-length = 120

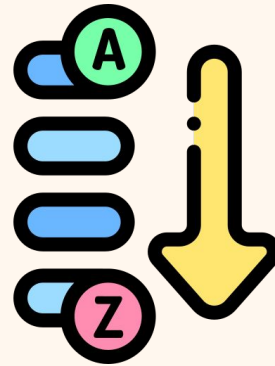
["pyproject.toml".tool.poetry.dev-dependencies]
pylint = "*"

["setup.cfg".flake8]
ignore = "D107,D202,D203,D401"
max-line-length = 120
inline-quotes = "double"

["setup.cfg".isort]
line_length = 120
multi_line_output = 3
include_trailing_comma = true
force_grid_wrap = 0
combine_as_imports = true
```

[сортировка TOML'ов](#)

★ 76



соцсеть из микросервисов!

CI/CD



Jenkins

★ 21.6k



GitLab

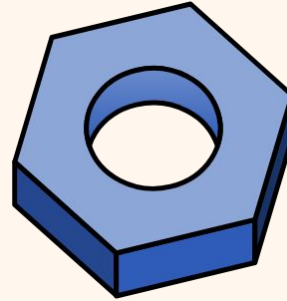
★ 23.4k

CI/CD



Jenkins

★ 21.6k



Buildbot

★ 5.1k



GitLab

★ 23.4k

Уголок здравого смысла



Jenkins

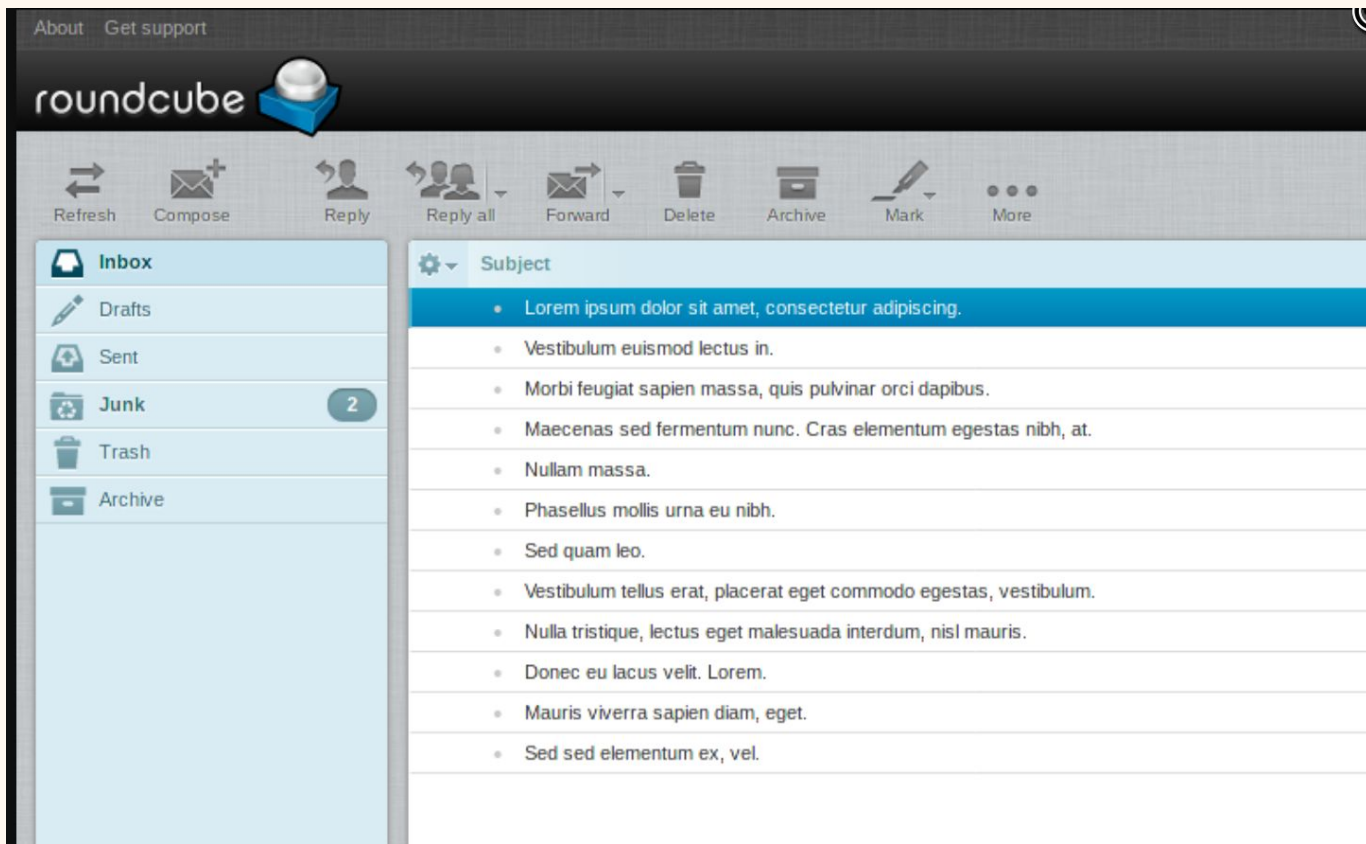
используйте



GitLab

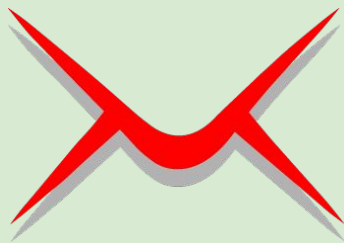
Почтовый сервер – [mailinabox](https://mailinabox.net/)

★ 12.5k



- postfix
- dovecot
- spamassassin
- opendkim
- ufw/fail2ban
- letsencrypt

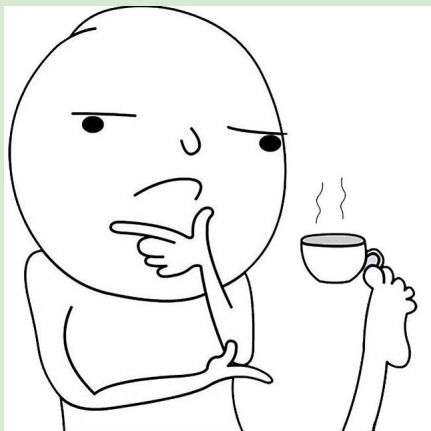
Уголок здорового ли смысла?



[iRedMail](#)

★ 1.2k

используйте




[Mailu](#)

★ 5k

уже на Python!

Запуск задач – Celery + Flower + Director

Celery Director 

Workflows

Status	Name	Date ↓
●	example.ETL	2020-10-05T17:45:10
●	example.RANDOMS	2020-10-05T17:44:30
●	example.ETL	2020-10-05T17:43:49
●	example.RANDOMS	2020-10-05T17:43:30
●	example.ETL	2020-10-05T17:43:15
●	example.ETL	2020-10-05T17:43:11
●	example.ETL_IN_ERROR	2020-10-05T17:42:49
●	example.RANDOMS	2020-10-05T17:39:00
●	example.RANDOMS	2020-10-05T17:38:49
●	example.ETL	2020-10-05T17:38:39

Rows per page: 10 1-10 of 93 < >

Search

Search by Date/ID

Filter by workflow

All example.ETL
example.RANDOMS
example.ETL_IN_ERROR

Filter by status

Select the status

Альтернативный мир

TASKiQ  384

```
1. from taskiq_nats import JetStreamBroker
2.
3. broker = JetStreamBroker("nats://localhost:4222",
4. queue="my_queue2")
5.
6. @broker.task
7. async def my_task(a: int, b: int) -> None:
8.     print("AB", a + b)
9.
10. async def main():
11.     await broker.startup()
12.     await my_task.kiq(1, 2)
13.     await broker.shutdown()
```

Procrastinate  608

```
1. import procrastinate
2.
3. app = procrastinate.App(
4.     connector=procrastinate.AiopgConnector())
5.
6. @app.task(queue="sums")
7. def sum(a, b):
8.     with open("myfile", "w") as f:
9.         f.write(str(a + b))
10.
11. with app.open():
12.     sum.defer(a=3, b=5)
13.     # ...
14.     app.run_worker(queues=["sums"])
```

Запуск задач - schedule



11.2k

```
1. import schedule
2. import time
3.
4. def job():
5.     print("I'm working...")
6.
7. schedule.every(10).minutes.do(job)
8. schedule.every().hour.do(job)
9. schedule.every().day.at("10:30").do(job)
10. schedule.every().monday.do(job)
11. schedule.every().wednesday.at("13:15").do(job)
12. schedule.every().minute.at(":17").do(job)
13.
14. while True:
15.     schedule.run_pending()
16.     time.sleep(1)
```

Дедовский способ – простой советский doit

★ 1.7k

```
def task_imports():
    """find imports from a python module"""
    return {
        'file_dep': ['projects/requests/requests/models.py'],
        'targets': ['requests.models.deps'],
        'actions': ['python -m import_deps %(dependencies)s > %(targets)s'],
        'clean': True,
    }
```

```
$ doit imports
-- imports
```

```
$ doit clean
imports - removing file 'requests.models.deps'
```



Уголок здравого смысла

используйте



CELERY



22.5k



Apache

Airflow



32.4k

тем более на Python!

ChatOps - errbot и opsdroid



```
import re
from errbot import BotPlugin, re_botcmd

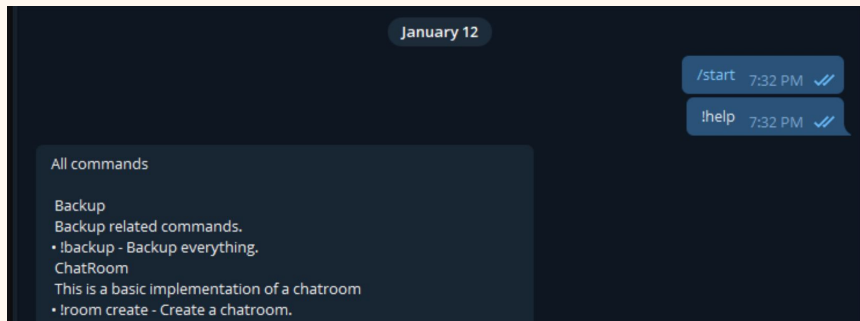
class CookieBot(BotPlugin):
    """A cookiemonster bot"""

    @re_botcmd(
        pattern=r"(\^| )cookies?( |$)",
        prefixed=False,
        flags=re.IGNORECASE)
    def listen_for_talk_of_cookies(self, msg, match):
        return "Somebody mentioned cookies? Om nom nom!"
```

```
from opsdroid.skill import Skill
from opsdroid.matchers import match_regex

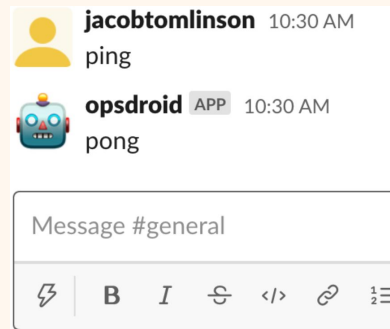
class PingSkill(Skill):

    @match_regex(r"ping")
    async def ping(self, event):
        await event.respond("pong")
```



```
connectors:
  slack:
    token: "MY API TOKEN"

skills:
  hello: {}
  ping:
    path: ~/opsdroid/myskill.py
```



Мониторинг - HE Graphite 5.8k

Installing in the Default Location

To install Graphite in the [default location](#), `/opt/graphite/`, simply execute `python setup.py install` as root in each of the project directories for Graphite-web, Carbon, Whisper, and Ceres.



Мониторинг - Glances

★ 23.8k

```
XP513-9333 (Ubuntu 22.04 64bit / Linux 5.15.0-71-generic) - IP 192.168.1.14/24 Pub 92.151. Europe/France/France Telecom - Orange Uptime: 6 days, 20:04:12

Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz - 1.80/2.02GHz CPU ↑ 100.0% idle 0.0% ctx_sw 3K MEM - 67.8% active 2.72G SWAP - 27.4% LOAD - 4core
CPU [|||||] user 98.3% irq 0.0% inter 2K total 7.30G inactl 2.67G total 1 min 1.90
MEM [|||||] 67.8% system 1.7% nice 0.0% sw_int 800 used 4.95G buffer 424M used 5 min 1.12
SWAP [|||||] 27.4% lowat 0.0% steal 0.0% free 2.35G cached 1.82M free 5.47G 15 min 0.89

NETWORK Rx/s Tx/s CONTAINERS 3 sorted by CPU consumption
Docker0 0b 744b
lo 0b 64b
vethccdb0e6 0b 744b podman 8d0f1c783def frosty bouman running 6 days 0.0 1.54M/7.30G 0b 0b 0b 0b top
wlp2s0 10kb 976b docker - portainer running yesterday 0.0 13.2M/7.30G 0b 0b 536b 0b /portainer
podman 8d0f1c783def 8d0f1c783def-infra running 6 days 0.0 456K/7.30G 0b 0b 0b 0b

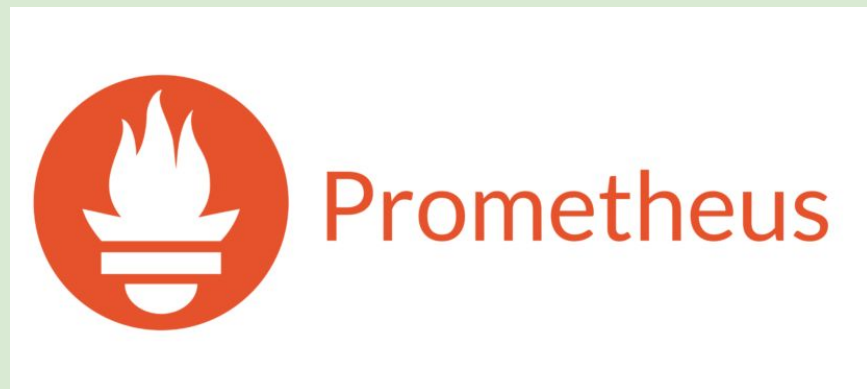
TCP CONNECTIONS
lIsten 39 TASKS 388 (1589 thr), 6 run, 316 slp, 66 oth Threads sorted automatically by CPU consumption
Initiated 0
Established 16 Dropbox 1 Up to date
Terminated 0 Python 7 CPU: 13.4% | MEM: 5.4%
Tracked 140/262144 Systemd 6 Services loaded: 383 active; 382 failed; 1

DefaultGateway 25ms CPU% MEM% VIRT RES PID USER TIME+ THR NI S R/s W/s Command ('e' to pin | 'k' to kill)
DISK I/O R/s W/s
dn-0 25K 490K 82.7 0.0 3.62M 112K 167177 nicolargo 0:11 1 0 R 0 0 stress --cpu 4 -t 15
dn-1 0 0 80.6 0.0 3.62M 112K 167176 nicolargo 0:10 1 0 R 0 0 stress --cpu 4 -t 15
sda 25K 490K 13.9 0.7 567M 52.3M 166557 nicolargo 0:19 6 0 R 0 0 python -m glances
sda1 0 0 10.6 0.6 608M 42.6M 166705 nicolargo 0:02 4 0 S 0 0 gnome-screenshot --gapplication-service
sda2 0 0 10.4 3.5 5.49G 259M 3927 nicolargo 39:30 19 0 S 0 0 gnome-shell
sda5 25K 490K 7.5 1.3 969M 97.2M 11646 nicolargo 7:00 4 0 S 0 0 python3 /usr/bin/terminator
0.9 0.6 1.35G 46.8M 166568 nicolargo 0:01 10 0 S 0 7K podman --log-level=info system service
FILE SYS Used Total 0.7 5.8 20.4G 430M 10541 nicolargo 1h25:25 147 0 S 0 2K firefox
/ 206G 227G 0.5 0.1 756M 8.73M 1727 root 1:20 10 0 S 0 ? ? multipassd --verbosity debug --logger platform
/./common/host-hunspell 206G 227G 0.5 0.1 309M 7.99M 4086 nicolargo 2:42 3 0 S 0 0 dbus-daemon --panel disable
/zsfpool (zsfpool) 128K 40.0M 0.2 7.8 3.46G 586M 11043 nicolargo 13:06 20 0 S 0 0 firefox -contentproc -childID 6 -isForBrowser -prefsLen 38436 -prefMapSize 241898 -jsInitLen 240056
FOLDERS 0.2 4.1 2.93G 303M 10770 nicolargo 7:34 22 0 S 0 0 firefox -contentproc -childID 2 -isForBrowser -prefsLen 27003 -prefMapSize 241898 -jsInitLen 240056
/home/nicolargo/Videos 26.2G 0.2 3.4 2.73G 251M 10774 nicolargo 2:44 22 0 S 0 0 firefox -contentproc -childID 3 -isForBrowser -prefsLen 27003 -prefMapSize 241898 -jsInitLen 240056
0.2 2.7 1.10T 200M 59195 nicolargo 19:58 16 0 S 0 0 code --type=renderer --crashpad-handler-pid=59145 --enable-crash-reporter=721e05a9-6035-4dcb-bd58-6
0.2 1.6 3.00G 119M 4288 nicolargo 3:32 94 0 S 0 0 dropbox
0.2 1.3 2.47G 94.1M 10790 nicolargo 0:47 22 0 S 0 0 firefox -contentproc -childID 5 -isForBrowser -prefsLen 27003 -prefMapSize 241898 -jsInitLen 240056
acpitzz 0 27C 0.2 0.2 1.37G 18.3M 1771 root 0:49 10 0 S 0 ? ? containerd
acpitzz 1 29C 0.2 0.1 2.02G 10.8M 3115 plex 2:00 26 0 S 0 ? ? Plex DLNA Server
Package Id 0 68C 0.2 0.1 335M 6.20M 4169 nicolargo 0:35 4 0 S 0 0 dbus-extension-gtk3
Core 0 64C 0.2 0.1 704M 4.55M 56119 root 0:16 12 0 S 0 ? ? containerd-shim-runc-v2 -namespace moby -id 3abd51c615968482d9ccff5af6c29f267fdda113ed60b75b432615
Core 1 68C 0.2 0.0 7.13M 1.06M 12492 nicolargo 0:09 1 0 S 0 0 top
CPU 69C 0.2 0.0 2.72M 80K 1577 root 0:18 1 -20 S 0 ? ? atopacctd
SODINM 44C 0.2 0.0 0 0 14 root 0:44 1 0 I 0 ? ? [rcu_sched]
BAT BAT0 100%✓ 0.2 0.0 0 0 159829 root 0:02 1 0 I 0 ? ? [kworker/u16:21-flush-253:0]
0.2 0.0 0 0 163786 root 0:01 1 0 I 0 ? ? [kworker/0:1-events]
0.2 0.0 0 0 166556 root 0:00 1 0 I 0 ? ? [kworker/1:2-mm_percpu_wq]
0.2 0.0 0 0 166982 root 0:00 1 0 I 0 ? ? [kworker/3:3-events]
0.0 4.5 3.73G 338M 55857 root 8:22 47 0 S 0 ? ? dockerd -H fd:// --containerd=/run/containerd/containerd.sock
0.0 3.8 3.01G 286M 10778 nicolargo 15:01 24 0 S 0 0 firefox -contentproc -childID 4 -isForBrowser -prefsLen 27003 -prefMapSize 241898 -jsInitLen 240056
0.0 3.0 2.59G 221M 82905 nicolargo 1:52 21 0 S 0 0 firefox -contentproc -childID 676 -isForBrowser -prefsLen 32152 -prefMapSize 241898 -jsInitLen 2400

High CPU user mode
2023-05-15 09:05:53 CEST 2023-05-15 09:05:45 (ongoing) - CPU_TOTAL (Min:85.0 Mean:94.0 Max:100.0): stress, stress, stress
```

Уголок здравого смысла

используйте



★ 50.6k

Event sourcing



```
class Dog(Aggregate):
    @event('Registered')
    def __init__(self, name):
        self.name = name
        self.tricks = []

    @event('TrickAdded')
    def add_trick(self, trick):
        self.tricks.append(trick)
```

```
class DogSchool(Application):
    def register_dog(self, name):
        dog = Dog(name)
        self.save(dog)
        return dog.id

    def add_trick(self, dog_id, trick):
        dog = self.repository.get(dog_id)
        dog.add_trick(trick=trick)
        self.save(dog)

    def get_dog(self, dog_id):
        dog = self.repository.get(dog_id)
        return {
            'name': dog.name,
            'tricks': tuple(dog.tricks)
        }
```

```
dog_id = application.register_dog(name='Fido')
application.add_trick(dog_id, trick='roll over')
application.add_trick(dog_id, trick='fetch ball')
dog_details = application.get_dog(dog_id)
assert dog_details['name'] == 'Fido'
```

```
notifications = application.notification_log.select(start=1, limit=10)
assert len(notifications) == 3
assert notifications[0].id == 1
```

```
from flask import request
tape_cassette = S3TapeCassette('production-recordings', region='us-east-1', read_only=False)
tape_recorder = TapeRecorder(tape_cassette)
tape_recorder.enabled_recording()
```

```
class ServiceOperation(object):
```

```
    @tape_recorder.operation()
```

```
    def execute(self):
```

```
        data = self.get_request_data()
```

```
        result = self.do_something_with_input(data)
```

```
        storage_key = self.store_result(result)
```

```
        return storage_key
```

```
    @tape_recorder.intercept_input(alias='service_operation.get_request_data')
```

```
    def get_request_data(self):
```

```
        return request.data
```

```
    @tape_recorder.intercept_output(alias='service_operation.store_result')
```

```
    def store_result(self, result):
```

```
        result_key = self.put_result_in_mongo(result)
```

```
        return result_key
```

playback



```
tape_cassette = S3TapeCassette('production-recordings', region='us-east-1')
tape_recorder = TapeRecorder(tape_cassette)
```

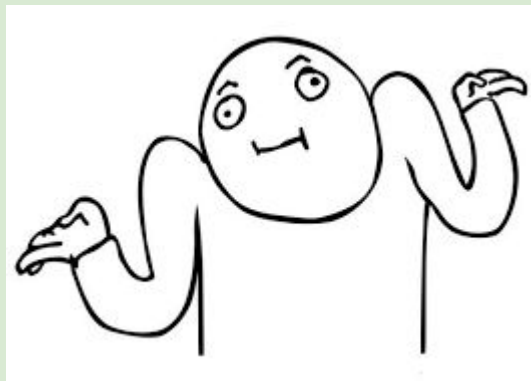
```
def playback_function(recording):
    """
    Given a recording, starts the execution of the recorded operation
    """
    operation_class = recording.get_metadata()[TapeRecorder.OPERATION_CLASS]
    return operation_class().execute()
```

```
# Will replay recorded operation, injecting and capturing needed data
# in all of the intercepted inputs and outputs
tape_recorder.play(recording_id, playback_function)
```



Уголок здравого смысла

используйте модуль [logging](#)



Виртуалки в клауде – [cloud-init](#)

★ 2.4k

Запускается автоматически при развертке образа в публичном или приватном облаке



Инфраструктура – PyInfra

★ 2.4k

```
from pyinfra.operations import apt, server

# Define some state - this operation will do nothing on subsequent runs
apt.packages(
    name="Ensure the vim apt package is installed",
    packages=["vim"],
    _sudo=True, # use sudo when installing the packages
)
```



Уголок здравого смысла

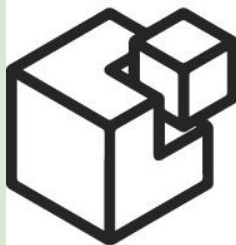
используйте



ANSIBLE



59.2k



SALTSTACK



13.6k

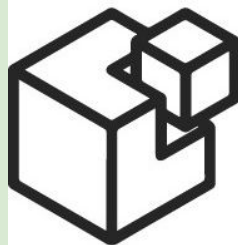
Уголок здравого смысла

используйте



ANSIBLE

★ 59.2k



SALTSTACK

уже на Python!

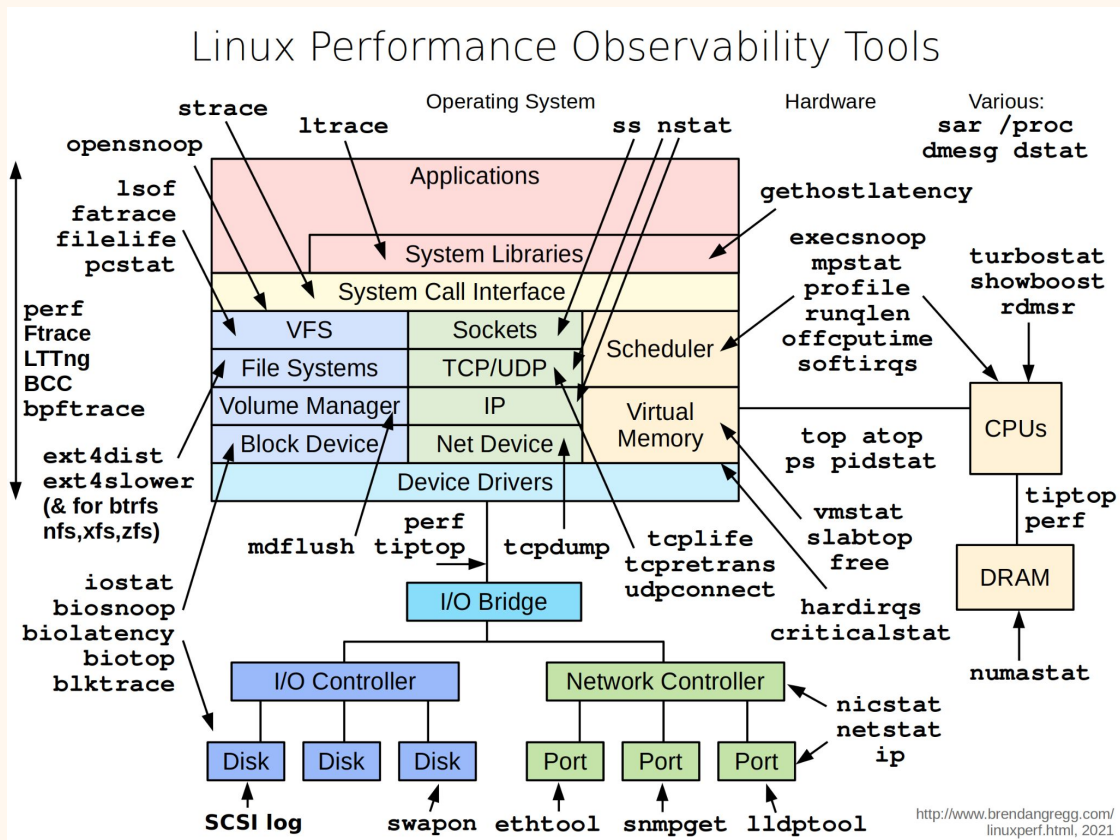
★ 13.6k

An abstract painting with a complex, layered composition. The background is a dense mix of colors, including deep blues, purples, oranges, reds, and pinks, with visible brushstrokes and textures. A semi-transparent, rounded rectangular box is centered horizontally across the middle of the image, containing the text "idkfa iddqd idclip" in a white, sans-serif font.

idkfa iddqd idclip

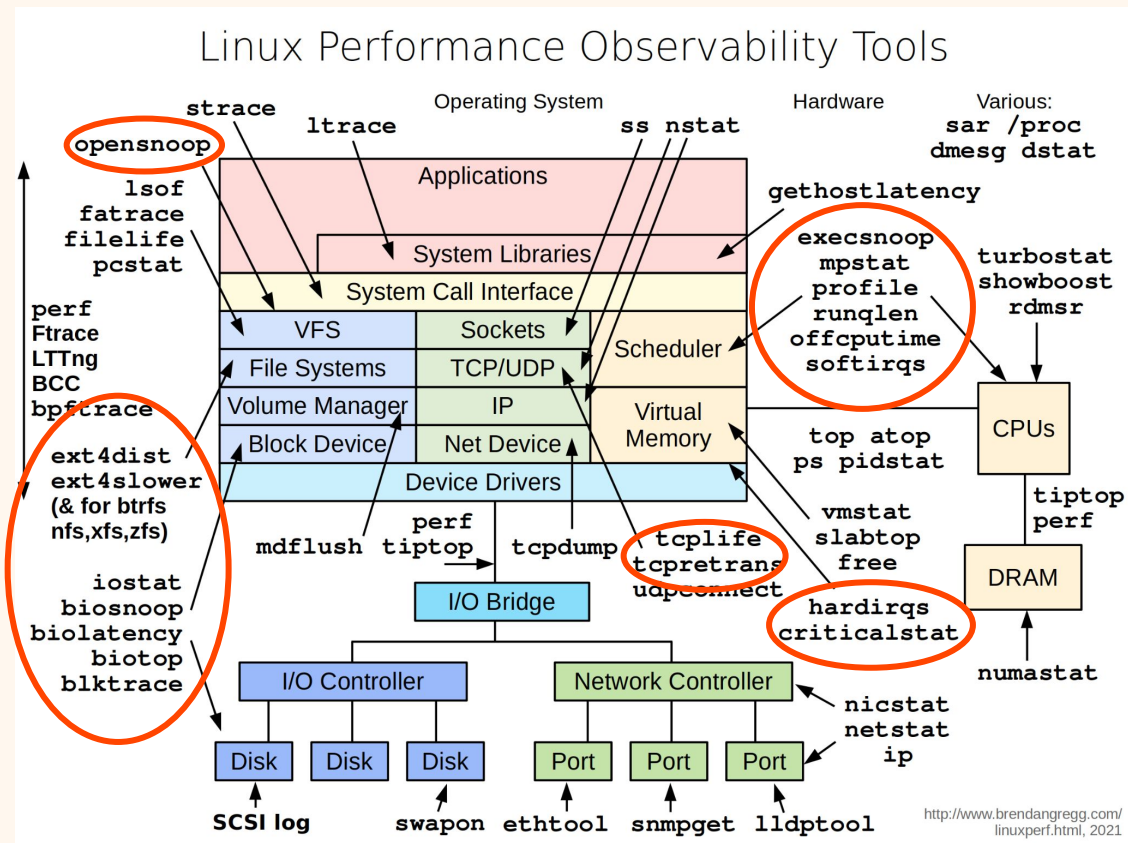
ВСС - режим бога

★ 18.3k



ВСС - режим бога

★ 18.3k



An abstract painting with a complex, layered composition. The background is a dense mix of colors including deep blues, purples, oranges, reds, and pinks, with visible brushstrokes and splatters. A semi-transparent, rounded rectangular box is centered horizontally across the middle of the image, containing the text "Write your own X" in a white, serif font.

Write your own X

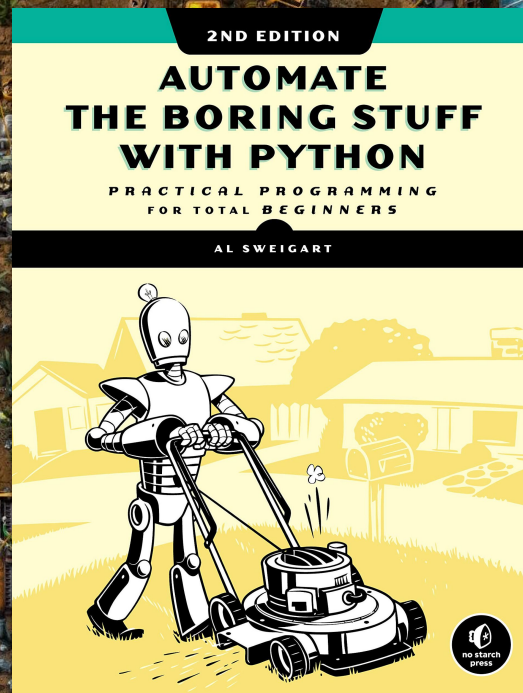
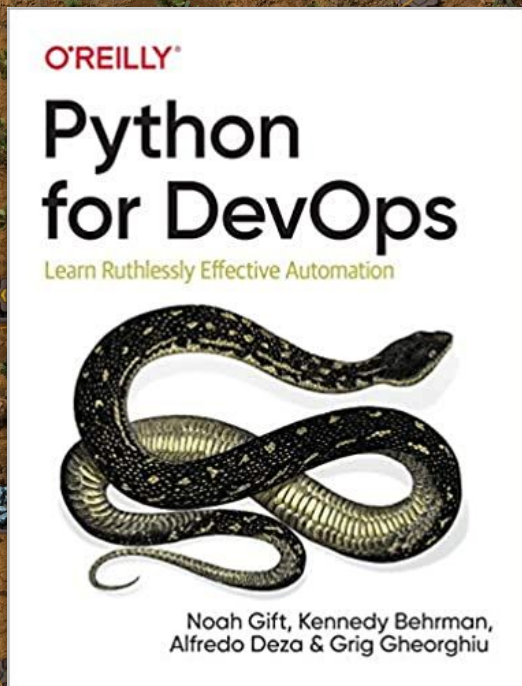
Architecture of Open Source Applications



[Написать свое](#)



Почитать



[Factorio](#)

Factorio-draftsman

```
from draftsman.blueprintable import Blueprint, BlueprintBook
from draftsman.constants import Direction
from draftsman.entity import ConstantCombinator
```

```
blueprint = Blueprint()
blueprint.label = "Example"
blueprint.description = "A blueprint for the readme."
blueprint.version = (1, 0) # 1.0
```

```
# Flexible ways to specify entities
blueprint.entities.append(
    "decider-combinator",
    id="clock",
    tile_position=[0, 3],
    direction=Direction.EAST,
    control_behavior={
        "decider_conditions": {
            "first_signal": "signal-red",
            "comparator": "<=",
            "constant": 60,
            "output_signal": "signal-red",
            "copy_count_from_input": True,
        }
    },
)
```



Python-sc2

★ 401

```
from sc2 import maps
from sc2.player import Bot, Computer
from sc2.main import run_game
from sc2.data import Race, Difficulty
from sc2.bot_ai import BotAI

class WorkerRushBot(BotAI):
    async def on_step(self, iteration: int):
        if iteration == 0:
            for worker in self.workers:

worker.attack(self.enemy_start_locations[0])

run_game(maps.get("Abyssal Reef LE"), [
    Bot(Race.Zerg, WorkerRushBot()),
    Computer(Race.Protoss, Difficulty.Medium)
], realtime=True)
```



Вопросы?

