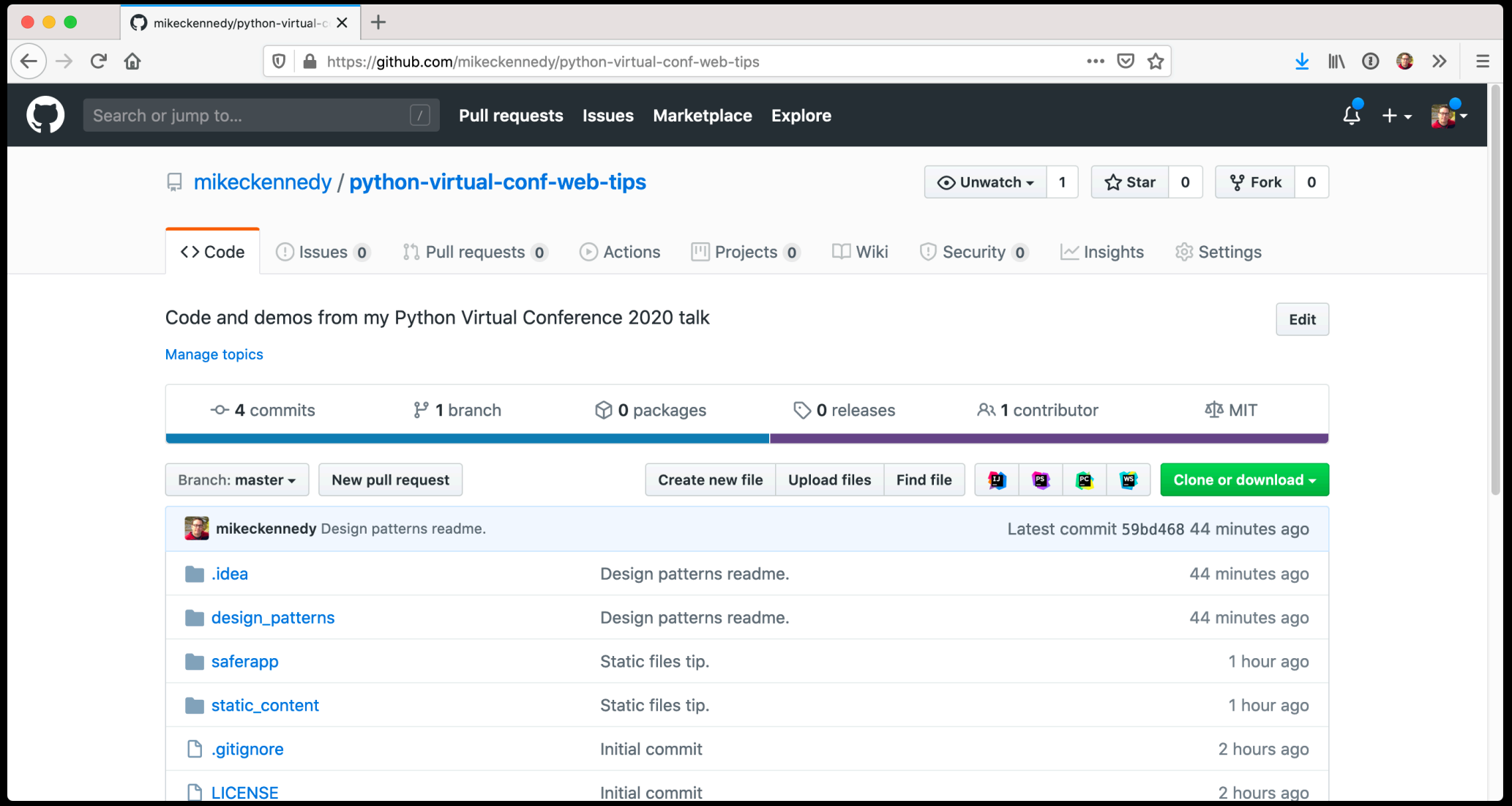
10 Tips Learned from Running A Python [Pyramid] Web App in Production

Michael Kennedy
@mkennedy

Getting the source code



https://github.com/mikeckennedy/python-virtual-conf-web-tips


TIP

Most tips work for most frameworks

Secure.py

Secure.py

OWASP Foundation | Open Sou

https://owasp.org

Register now for Virtual AppSec Days Summer of Security. Classes online June, July & August

OWASP

PROJECTS CHAPTERS EVENTS ABOUT

Search OWASP.org

Donate

Join

Who is the OWASP Foundation?

The Open Web Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software. Through community-led open source software projects, hundreds of local chapters worldwide, tens of thousands of members, and leading educational and training conferences, the OWASP Foundation is the source for developers and technologists to secure the web.

- Tools and Resources
- Community and Networking
- Education & Training

For nearly two decades corporations, foundations, developers, and volunteers have supported the OWASP Foundation and its work. Donate, Join, or become a Corporate Member today.

Integrating secure.py

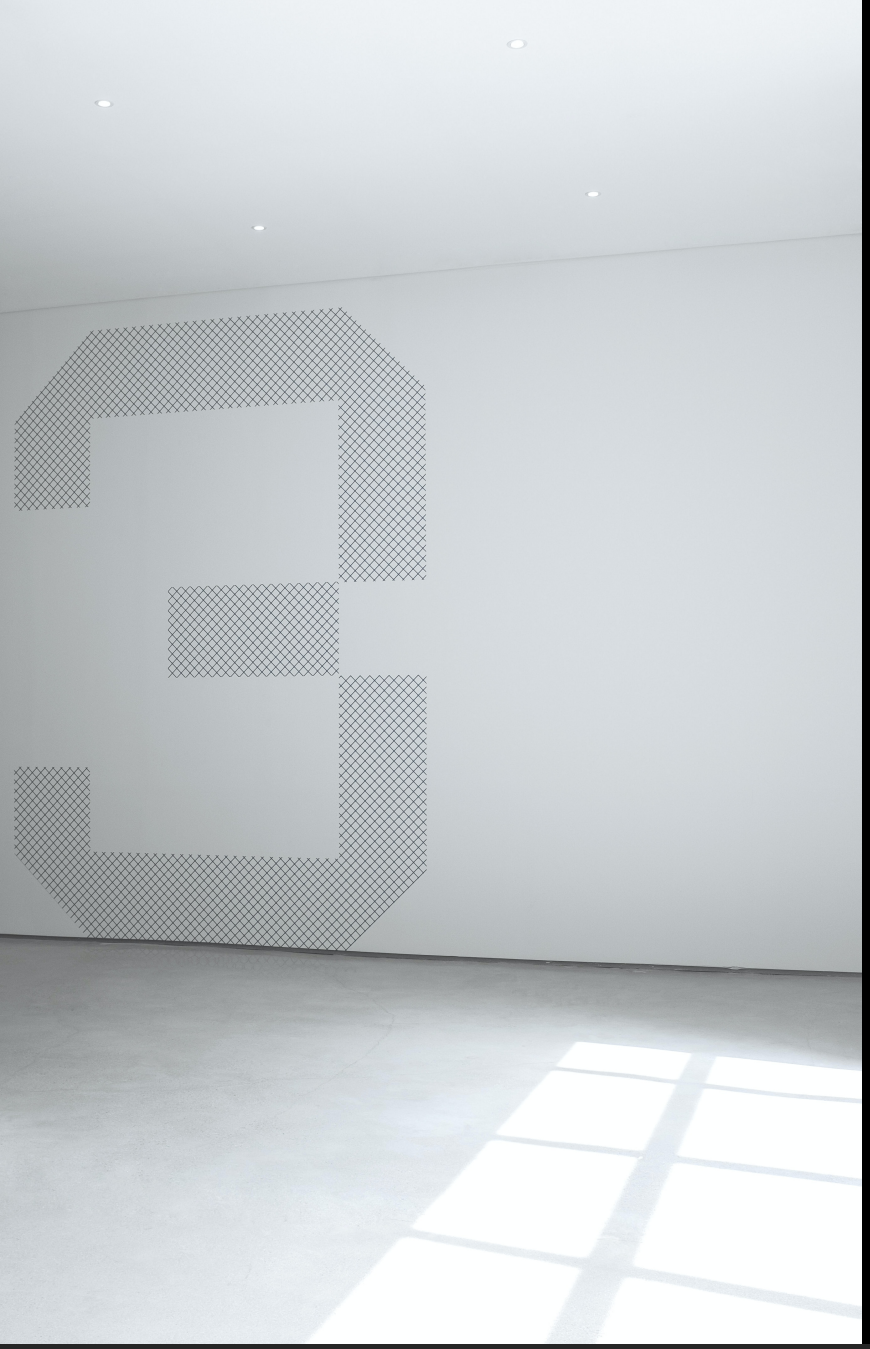
```
def set_secure_headers(handler, registry):
    import secure
    secure_headers = secure.SecureHeaders()

    def tween(request):
        response = handler(request)
        secure_headers.pyramid(response)
        return response

    return tween

def add_secure_headers(config):
    tween_name = 'saferapp.infrastructure.secure_tween.set_secure_headers'
    config.add_tween(tween_name)
```

https://secure.readthedocs.io/en/latest/frameworks.html

Static content out of nginx

site.nginx static settings

```
{
  # …

  location /static {
     alias /apps/saferapp/saferapp/static/;
     expires 365d; # <- Yes, one year!
     # We ensure the URLs include a hash of the actual file for images, css, js, etc
     # e.g /static/all.min.css?cache_id=f8b1d8771478c8623969b754682134fb
  }

}
```



Let's Encrypt

Steps for Let's Encrypt

```
# Update and set the dependencies
$ sudo apt-get update
$ sudo apt-get install software-properties-common
$ sudo add-apt-repository universe
$ sudo apt-get update

# Install certbot
$ sudo apt-get install certbot python3-certbot-nginx

# Detect nginx setups, find domains, create SSL setup
$ sudo certbot --nginx
```

https://certbot.eff.org/lets-encrypt/ubuntufocal-nginx

Design Patterns

Design Patterns

Pyramid: https://github.com/talkpython/data-driven-web-apps-with-pyramid-and-sqlalchemy
Flask: https://github.com/talkpython/data-driven-web-apps-with-flask

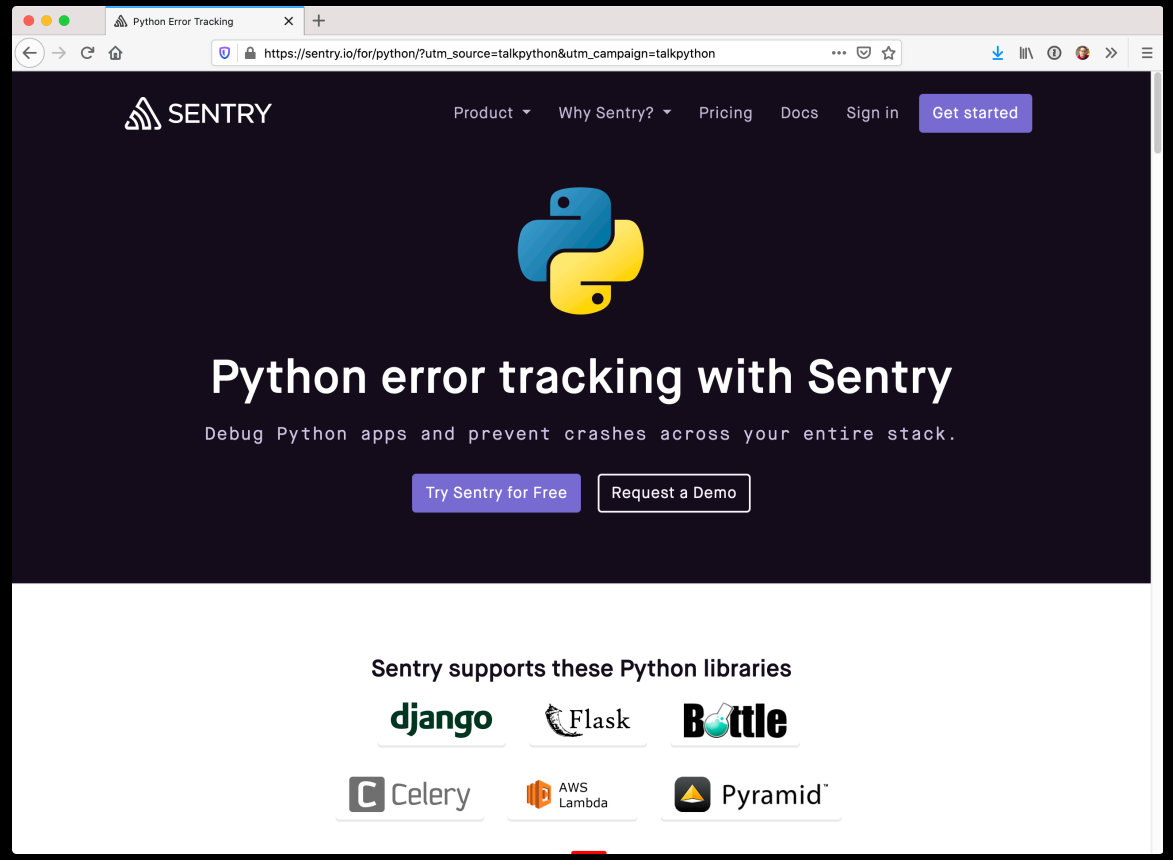
Patterns:

- ViewModels
- Services
- Strutured folders (view_module.view_name)

Error logging and reporting

Error logging and reporting

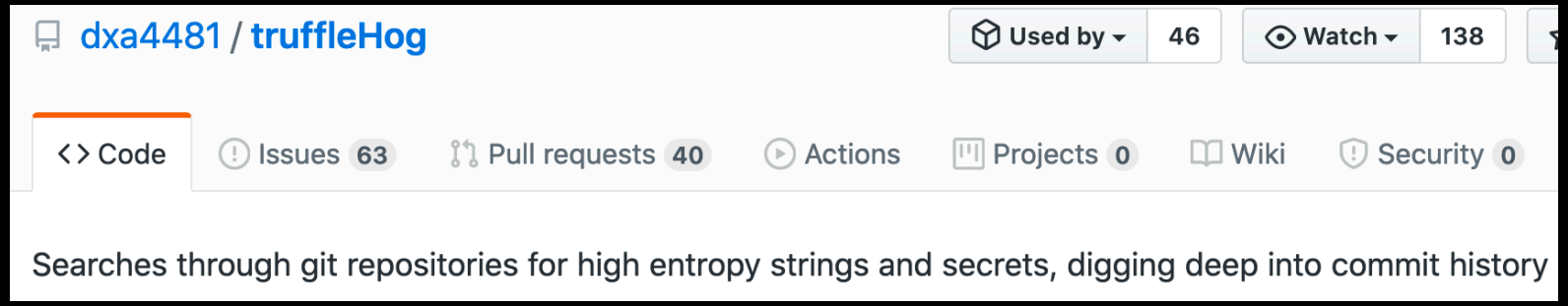
logbook.readthedocs.io/en/stable/

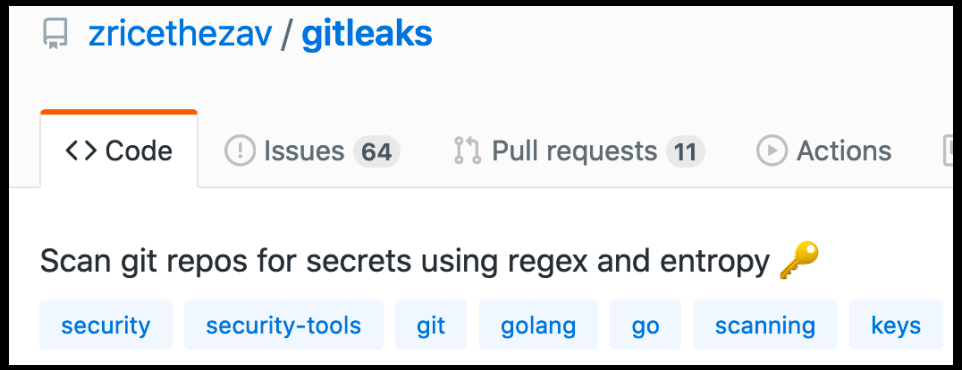


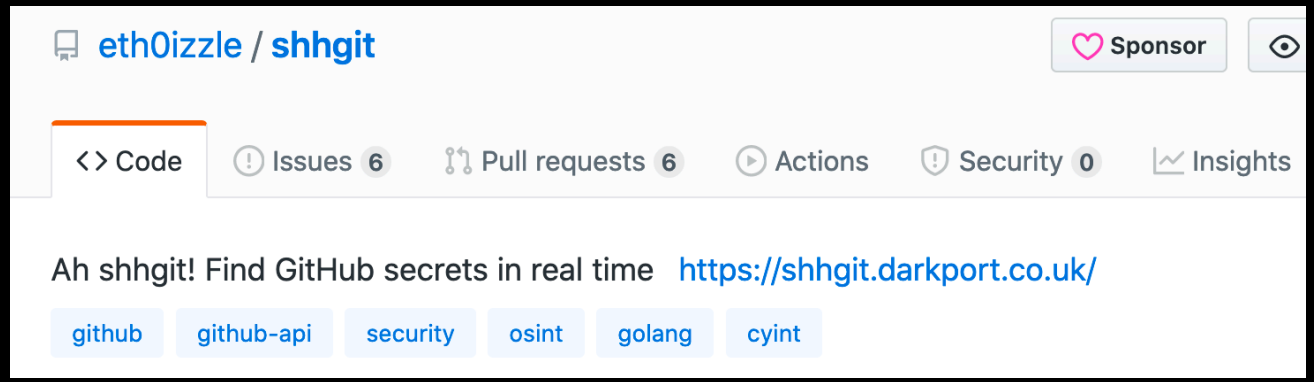
talkpython.fm/sentry

[Not] storing secrets

[Not] storing secrets







Shhgit finds secrets and sensitive files across GitHub code and Gists committed in near real-time by listening to the GitHub Events API.

[Not] storing secrets

First of all, security is about layers…

1. Store them in environment variables (or non-committed files).
2. Encrypt the secrets (store the key elsewhere, use **cryptography.fernet**)
3. Use key vaults (e.g. https://azure.microsoft.com/en-us/services/key-vault/)

Calvin Hendryx-Parker has some good tips on using 1Password automation:
https://sixfeetup.com/blog/managing-secrets-and-your-environment-with-1password

direnv can make this more automatic:
https://direnv.net/

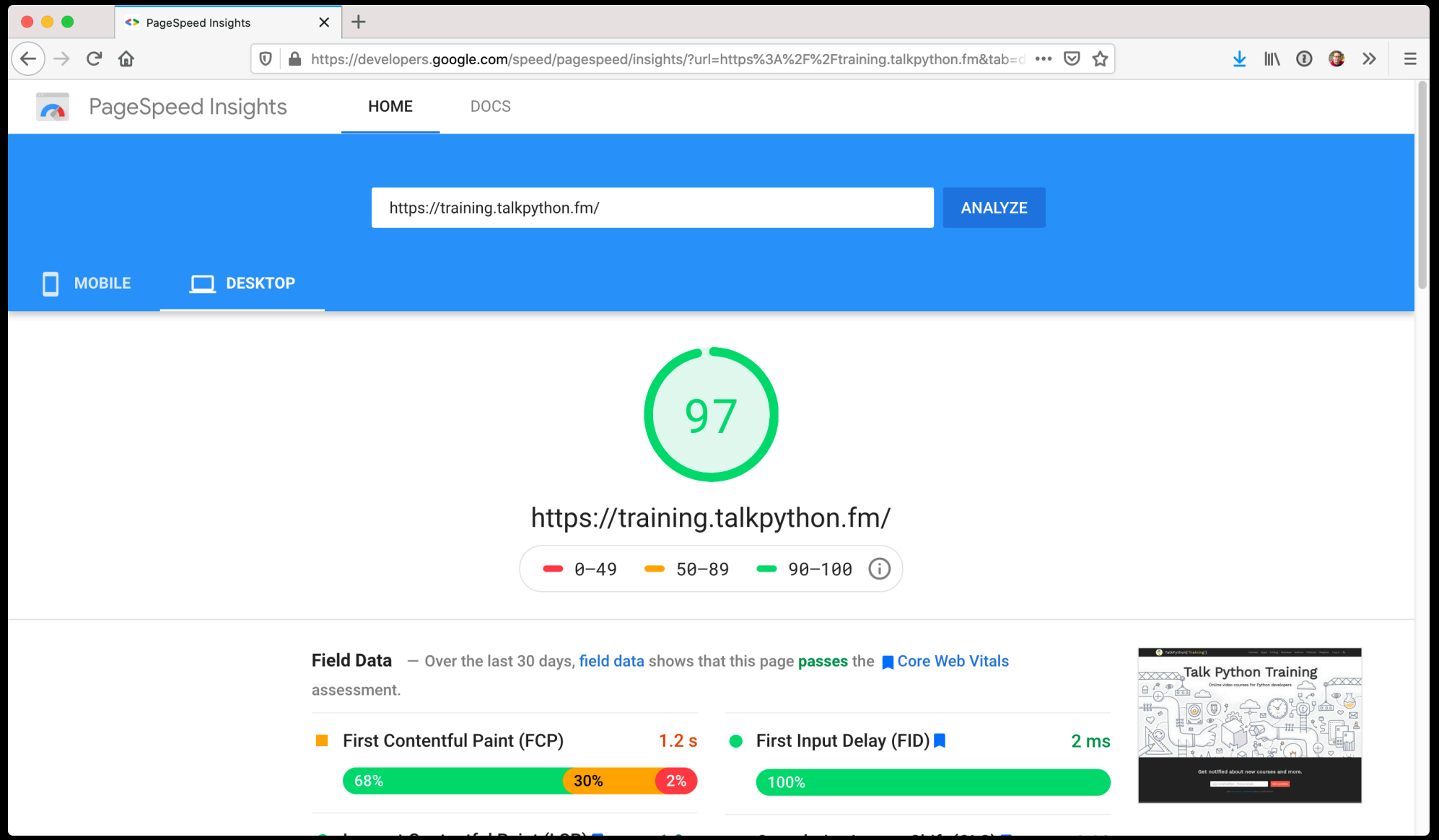
80/20 testing with sitemaps

80/20 testing with sitemaps

```
class SiteMapTests(TestCase):
    app = None

    def setUp(self):
        self.app = self.get_or_create_web_app()

    def get_sitemap_text(self):
        res = self.app.get("/sitemap.xml")
        return res.text.replace(
          'xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"', '')

    def test_site_mapped_urls(self):
        text = self.get_sitemap_text()
        x = xml.etree.ElementTree.fromstring(text)
        urls = [
            href.text.strip().replace('https://yourdomain.com', '')
            for href in list(x.findall('url/loc'))
        ]
        print(f'Testing {len(urls)} urls from sitemap...', flush=True)

        for url in urls:
            print('Testing url at ' + url)
            self.app.get(url, status=200)
```

Site speed

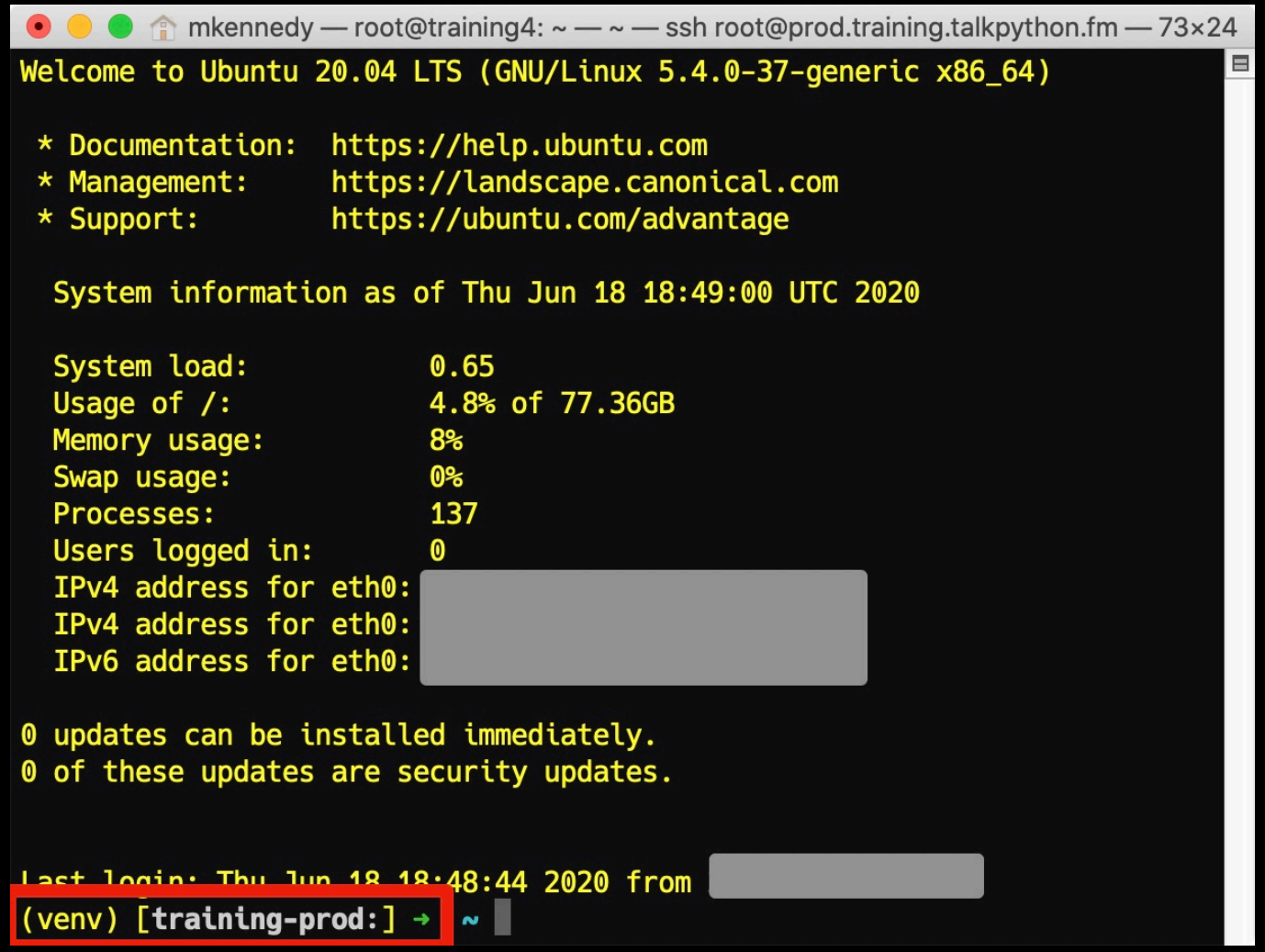
Site speed



https://developers.google.com/speed/pagespeed/insights/

Activate a venv at server login

Activate a venv at server login



Dependencies

Dependencies

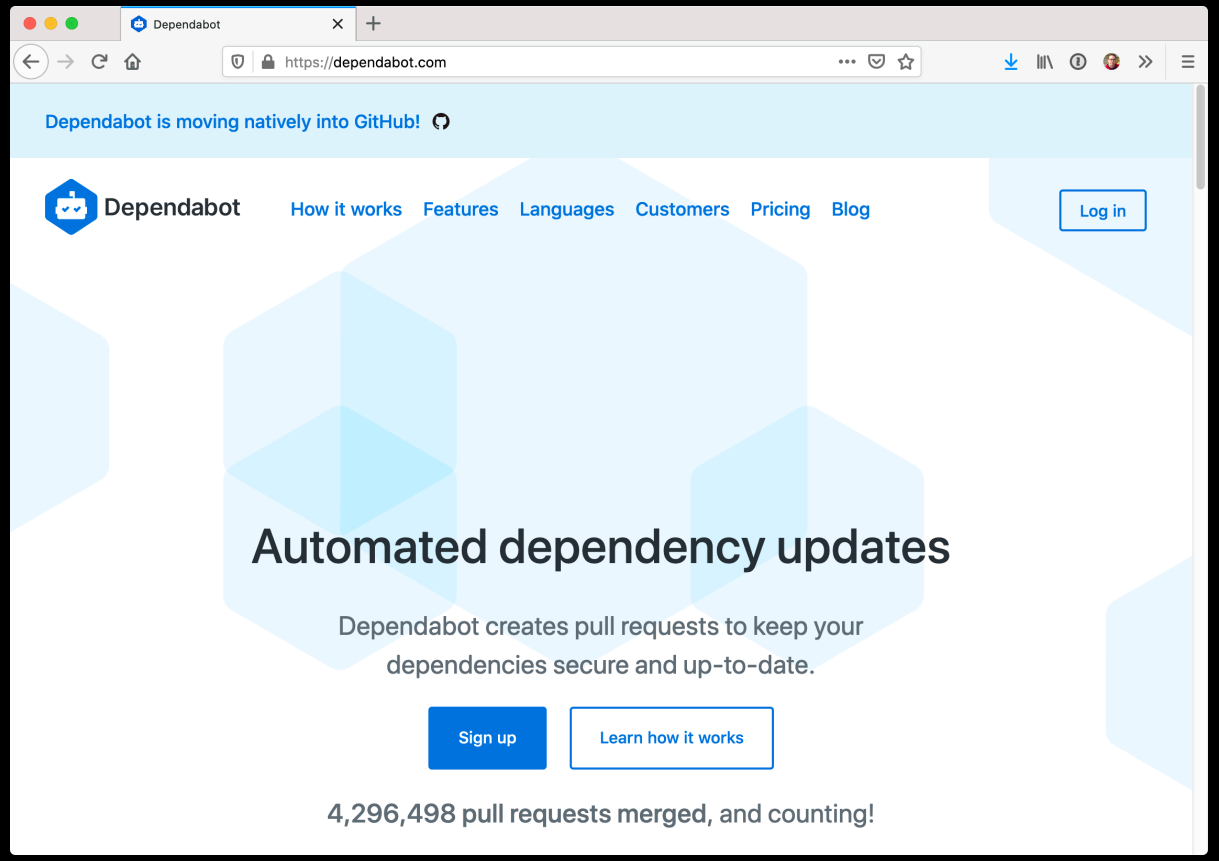
No big deal: `pip install pyramid` gets me the latest
Yes, unless it's already installed: `Requirement already satisfied`

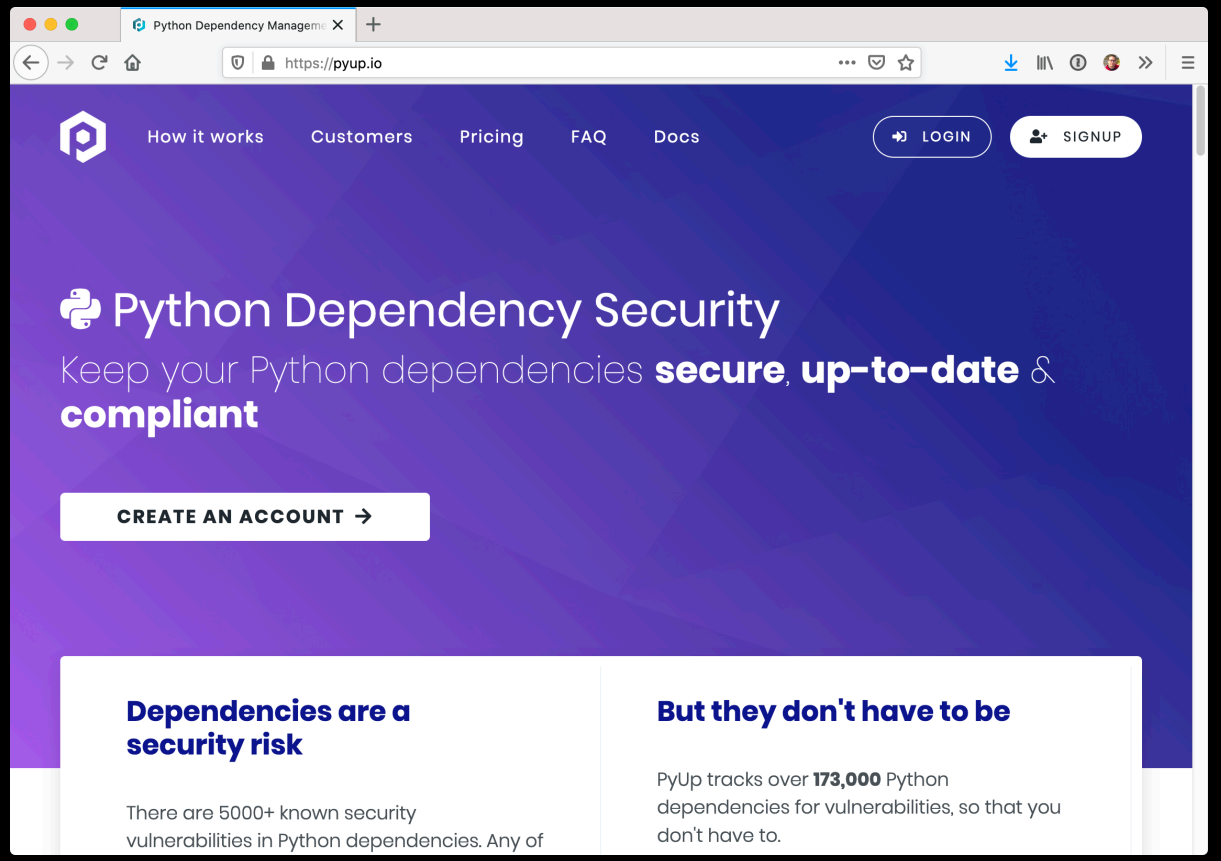
No big deal: `pip install pyramid --upgrade` gets me the latest
Yes, it's updated! But what about `venusian`, `WebOb`, etc.?

What about security issues that need attention right away?

No big deal: If you pin your version, GitHub will give security updates
Yes, but what about run of the mill updates? You'll be stuck in the past!

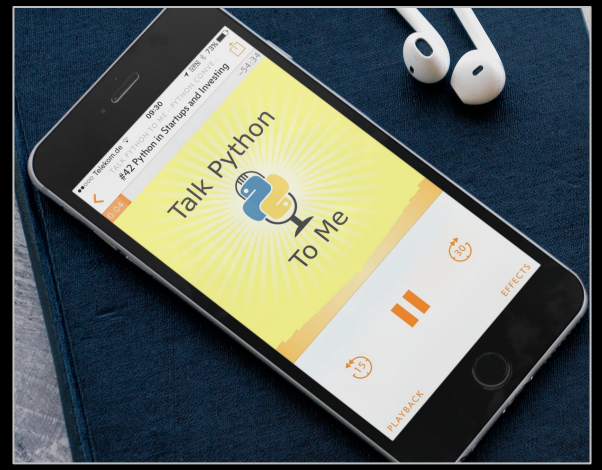
Dependencies





Thank you and where to get more

@mkennedy



Talk Python To Me Podcast

Python Bytes Podcast

Talk Python Training

https://github.com/mikeckennedy/python-virtual-conf-web-tips