

# IL FANTASTICO MONDO DEI MAKEFILE

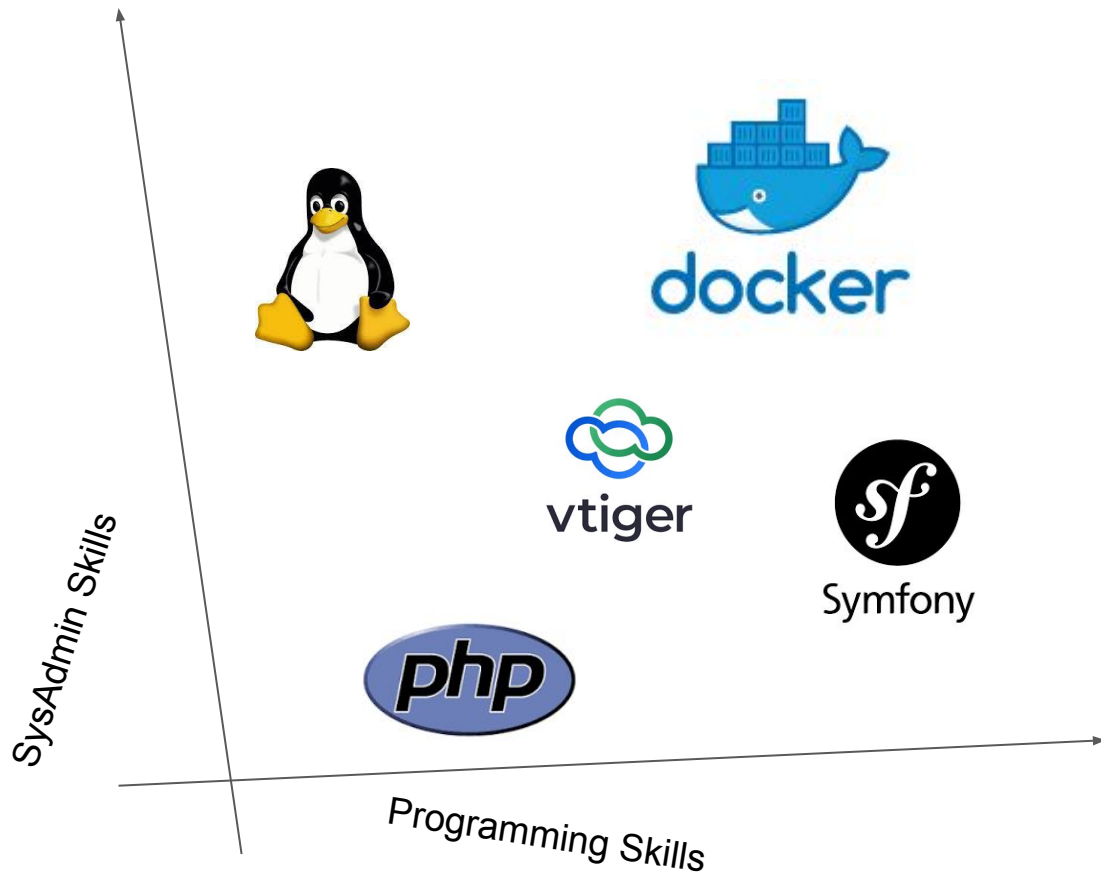


Ing. Francesco Bianco (he/him)

Open source CRM veteran  
DevOps vulgariser

<https://yafb.net/>

<https://github.com/javanile>



Nato nel 1981

Libera Professione dal 2009

Due splendidi figli e una  
moglie che mi sopporta

*"IBM nel 1981 lancia sul mercato  
il primo PC. Il mitico 5150 con  
processore 8088"*

**\$ make**





# Stuart Feldman

Ideatore e realizzatore del Fortran 77

Membro del gruppo che creò UNIX ai "Bell Labs"

Intuisce e crea il programma

**MAKE**

Nato nel 1946

Nell'aprile del 1976

Premiato nel 2003

# ACM Software System Award



Cos'è un

# MAKEFILE ?

```
target1:
    command1 "Hello World!"
    command2 --option1=hello --option2=world
    command3 run

target2:
    command1 "Sam Sepiol's page"
    command2 --option1=hello --option2=world
    command3 clean
```

```
~/my-project$ make target1
```

**meno target**

**più task**

- > Memorizzare i compiti
- > **Carico cognitivo (Cognitive load)**
- > Dev/Ops tasks

init

deploy

install

start

release

lock

stop

clean

debug

# Yet Another Task Runner



C/C++ 🤔

ma non solo...



Tell me a story

# Come si installavano le dipendenze nel 2000



`make install`



`unzip libs.zip`



`setuptools`  
`EasyInstall`



# Come si installavano le dipendenze nel 2005



`make install`



```
<script  
  src="jquery.js">  
</script>
```



`pip install`



`mvn install`

# Come si installavano le dipendenze nel 2012



`make install`



`bower install`



`pip install`



`gradle build`

# Come si installavano le dipendenze nel 2014



`make install`



`npm install`



`pip3 install`



`gradle build`

# Come si installavano le dipendenze nel 2015



`make install`



`grunt install`

# Come si installavano le dipendenze nel 2016



`make install`



`gulp install`

# Come si installavano le dipendenze nel 2017



`make install`



`yarn install`



# Come si installavano le dipendenze nel 2018



`make install`



`pnpm install`



# MAKEFILE

#MENO\_CARICO\_COGNITIVO più #BIRRA 

```
## I'm a Makefile  
## for JavaScript
```

```
install:
```

```
## Year 2012
```

```
#bower install
```

```
## Year 2014
```

```
#npm install
```

```
## Year 2016
```

```
yarn install
```

```
~/cheat-js$ make install
```

er

Angle: \_

dde



0>Score<0

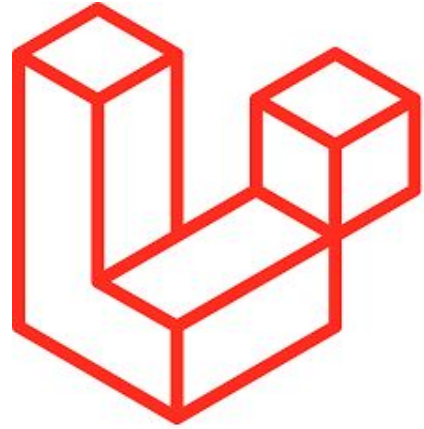
siamo liberi di pensare che:

# MAKEFILE

*non è un task runner,  
ma uno strumento utile  
ad abbassare  
il carico cognitivo  
di un progetto software*

Tiriamo le somme sui

# MAKEFILE



```
entity:
  php artisan make:controller $(name)Controller \
    --resource --model=$(name) --api --verbose

start:
  php artisan serve --quiet --no-reload \
    --host=$(laravel_host) --port=$(laravel_port)
```

```
~/my-laravel-app$ make entity name=Invoice
```

```
~/my-laravel-app$ make start
```

Tiriamo le somme sui

# MAKEFILE



start:

```
docker compose up -d --env-file=.env.docker \  
--remove-orphans --build --force-recreate
```

push:

```
git commit -am "Push to registry" && git push  
docker build -t my-docker-demo/server:latest .  
docker push my-docker-demo/server:latest
```

```
~/my-docker-demo$ make start
```

```
~/my-docker-demo$ make push
```

Tiriamo le somme sui

# MAKEFILE



```
deploy:
    kubectl apply -f wordpress-site.yaml \
        --namespace=$(domain) --record --dry-run=client \
        --timeout=5m --prune --selector=app=$(domain)

deep-restart:
    kubectl rollout restart deployment --all
```

```
~/my-kube-iac$ make deploy domain=winecity.com
```

```
~/my-kube-iac$ make deep-restart
```



**FINE**

*Grazie a tutt<sup>☆</sup>*

---



**Francesco Bianco**