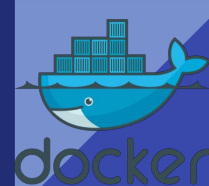


Build, test and release your python packages

Using DevPI, Docker and Jenkins

David Melamed

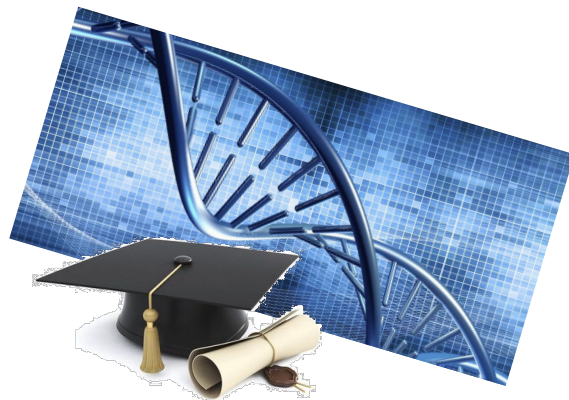


PyWeb #55 - 1st August 2016

A word about me



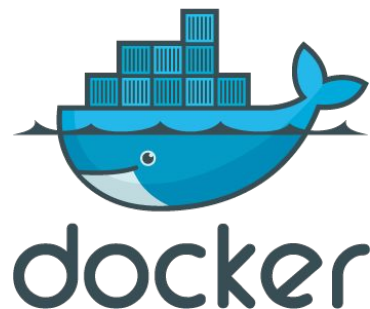
Moved to Israel 8 years ago



PhD in Bioinformatics



Senior Research Engineer



A word about CloudLock



Leader in the Cloud Access Security Broker (CASB) market

Founded: 2011

Corporate Headquarters: Waltham, Mass. (U.S.A.)

R&D Headquarters: Tel Aviv

Employees: 140 (30 in TLV)

Trusted by major brands:

Acquired by Cisco... **today!**



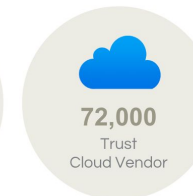
180K
APPS



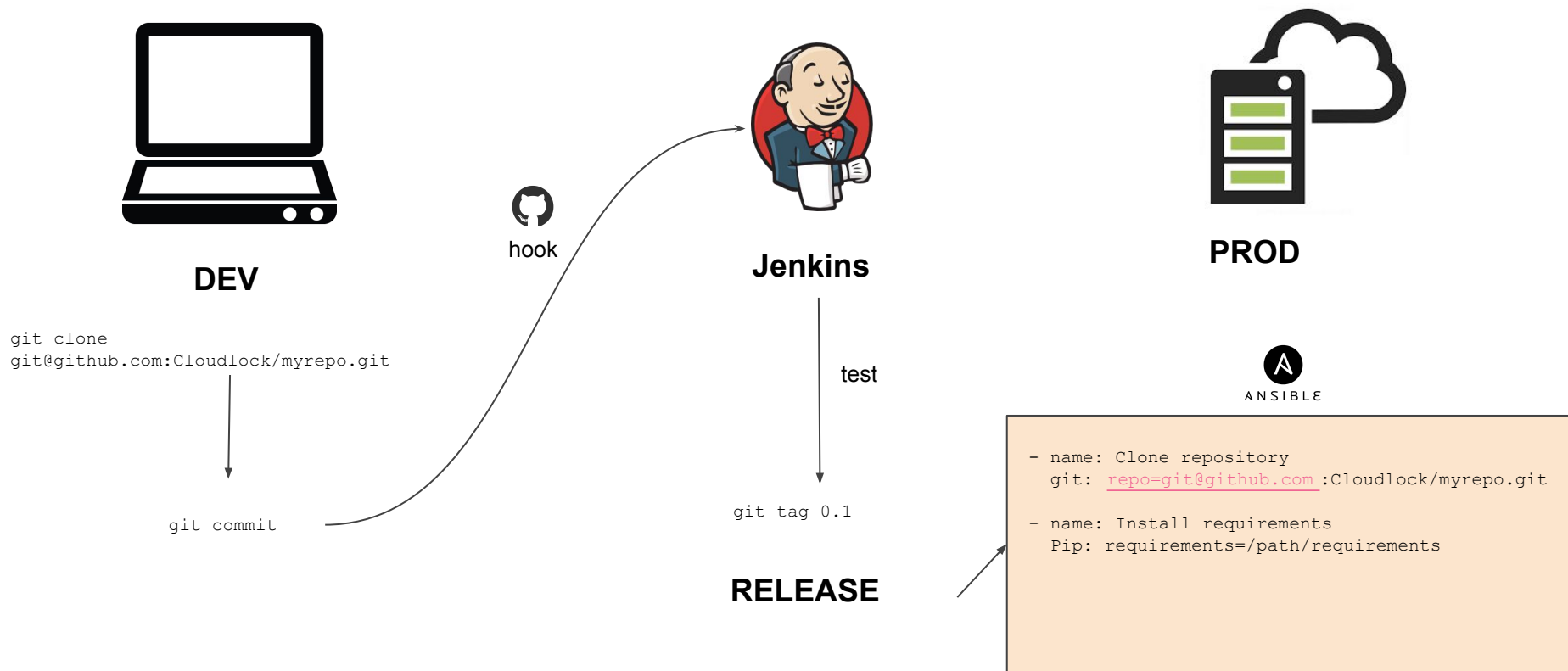
10 M
USERS



1 B / month
ACTIVITIES



CloudLock initial dev & release workflow



The problems

- PyPI may be not available
- No offline work
- Dependencies may change between dev & release
- Packages may be removed from repository
- Compiles source in production

What we would like

- Build immutable package
- Package code as wheels
- Being able to work offline
- Host private packages

DevPI main advantages

Local fast PyPI mirror

100% compatible with pip

Replication for offline use

Multiple indexes

Private indexes (dev, stage, prod)

Index inheritance

Web interface

UI for index navigation

Search for packages

Package documentation

The dev & release flow using DevPI and Jenkins



DevPI client main commands (1)

- Define the server

```
devpi use http://user:password@localhost/myindex/
```

- Create user

```
devpi user -c john email=john@abc.com password=12345
```

- Login as user

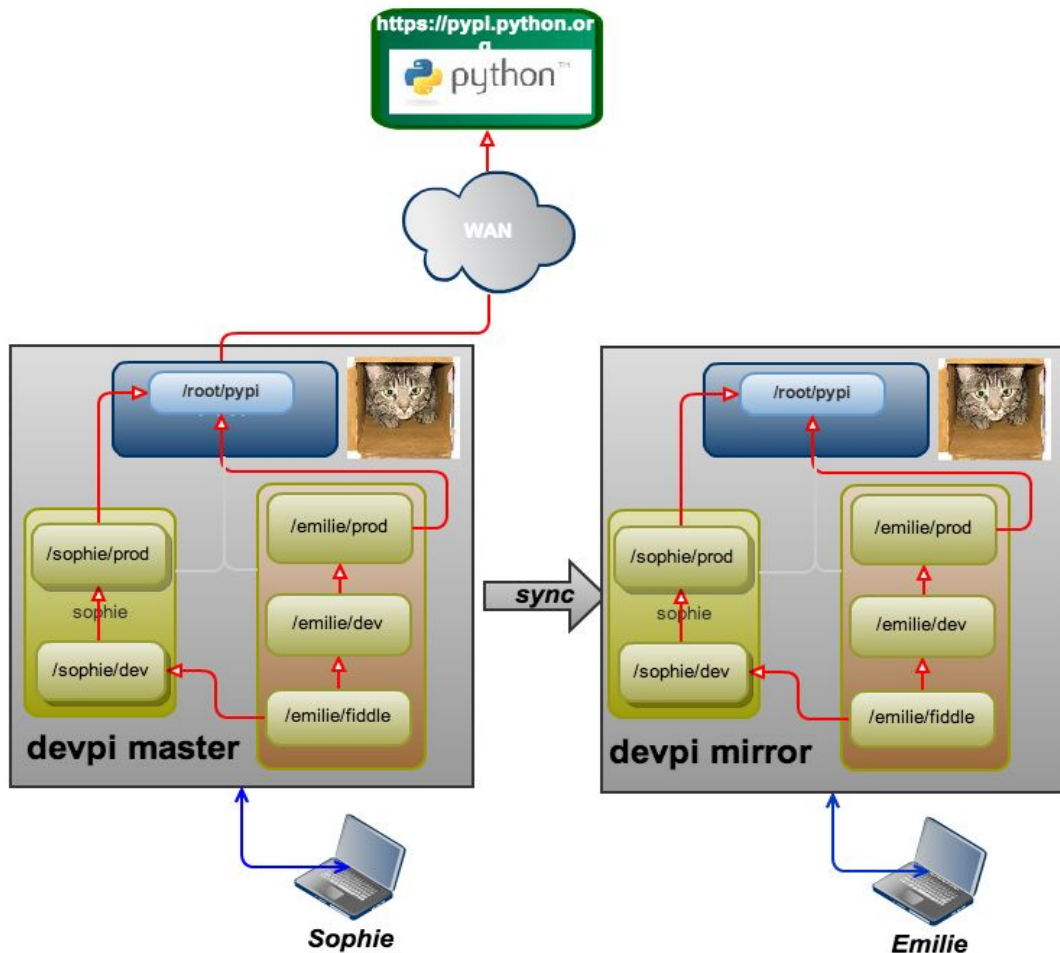
```
devpi login john --password=12345
```

- Upload package

```
devpi upload --formats=sdist,bdist_egg,bdist_wheel
```

DevPI indices

- Multiple private indices
- Index volatile / non-volatile
- Index inheritance



DevPI client main commands (2)

- List indices

```
devpi index -l
```

- Create an index

```
devpi index -c mycompany/prod volatile=False
```

```
devpi index -c john/dev bases=mycompany/prod
```

- Use an index

```
devpi use mycompany/prod
```

- Modify index properties

```
devpi index mycompany/prod acl_upload=jenkins \  
uploadtrigger_jenkins=http://jenkins-url \  
slack_hook=https://hooks.slack.com/services...
```

How to use devPI with pip

```
pip install requests -i http://user:password@devpi/myindex/+simple --trusted-host devpi
```

```
pip install requests -i https://user:password@devpi/myindex/+simple
```

PIP configuration

~/.pip/pip.conf

```
[global]
timeout = 60
index-url =
http://localhost/root/pypi/+simple/
download-cache = ~/.pip/cache
log-file = ~/.pip/pip.log
build = ~/.pip/build

use-wheel = true
find-links = ~/.pip-wheelhouse

[wheel]
wheel-dir = ~/.pip-wheelhouse
```

Environment variable

```
PIP_CONFIG_FILE=/etc/pip.conf

PIP_INDEX_URL=http://localhost/myuser/myidx/+simple/
PIP_TRUSTED_HOST=localhost
PIP_DISABLE_PIP_VERSION_CHECK=true
```

DevPI hooks & plugins

UI Theme

devpi-semantic-ui

Authentication

devpi-ldap

Upload hooks

devpi-jenkins

devpi-slack



Devpi

Search packages...



[🏠](#) > [root/pypi](#) > [flask](#) > **0.11.1**



flask 0.11.1

root/pypi

[🔗](#) Simple index [🔗](#) PyPI page

File	History
Flask-0.11.1-py2.py3-none-any.whl <small>? Python Wheel 📄 2.7</small>	
Flask-0.11.1.tar.gz <small>? Source</small>	

please refer to description on remote server <https://pypi.python.org/pypi/flask/0.11.1/>

About

[Search syntax](#)
[Documentation](#)

Versions

[devpi-semantic-ui-0.1.0](#)
[devpi-server-4.1.0](#)
[devpi-web-3.1.1](#)

Devpi

Server status ✔ ok

Install devPI with Docker

data

Container for the packages



devpi

The DevPI server itself

Exposes port 4040

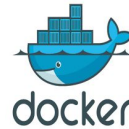


nginx

Basic auth for server access

- download packages
- UI

Exposes port 80



Install devPI with Docker - data container

data

```
FROM alpine:3.2

# Env vars
ENV PYTHONUNBUFFERED=1

# Server directory
RUN adduser -D devpi && mkdir -p /devpi/server && chown -R devpi:devpi /devpi

# Entry point, simply exit with status code 0
CMD ["/bin/true"]
```

Install devPI with Docker - devpi container

devpi

```
FROM alpine:3.2

# Env vars
ENV PYTHONUNBUFFERED=1

# Dependencies
RUN apk add --update python3 py-pip

RUN pip3 install devpi-server devpi-web devpi-semantic-ui

# Server directory
RUN adduser -D devpi && mkdir /devpi && chown -R devpi:devpi /devpi

# Copy static files
RUN cp -R /usr/lib/python3.4/site-packages/devpi_web/static /devpi/+static

# Expose volume
VOLUME ["/devpi/server"]

# Change to unprivileged user
USER devpi

# Entry point
EXPOSE 4040
CMD ["devpi-server", "--host", "0.0.0.0", "--port", "4040", "--serverdir", "/devpi/server", "--theme", "\
/usr/lib/python3.4/site-packages/devpi_semantic_ui", "--restrict-modify", "root"]
```

Install devPI with Docker (3)

nginx

```
FROM alpine:3.2

RUN apk add --update nginx

# Copy config file
COPY ["nginx.conf", "auth.conf", "htpasswd", "/etc/nginx/"]
RUN chown root:nginx /etc/nginx/htpasswd && chmod 660 /etc/nginx/htpasswd

# Change to unprivileged user
USER nginx

# Entry point
EXPOSE 80
CMD ["nginx"]
```

Run devpi with docker-compose

docker-compose.yml

```
version: '2'
services:
  data:
    build: ./data
  volumes:
    - /devpi/server

  devpi:
    restart: always
    build: ./devpi
    ports:
      - "4040:4040"
    volumes_from:
      - data

  nginx:
    restart: always
    build: ./nginx
    ports:
      - "80:8080"
    links:
      - devpi
    volumes_from:
      - data
```



want to work for us?

#infra software engineer

#qa engineer

#data scientist

#senior product manager

#senior software engineer

#ux designer

#cybersecurity analyst

#ui architect

#software engineer in test

Interested? Send an email to sapir@cloudlock.com