Управление окружениями.

Никита Казеев[1], Андрей Устюжанин[1,2]

[1]Yandex School of Data Analysis, [2]National Research University Higher School of Economics

Содержание

- Окружениями в python
- Docker

Окружения в Python

- virtualenv
  - `pip install virtualenv`
- conda
  - `http://conda.pydata.org/miniconda.html`

virtualenv

```
pip install virtualenv                                  # 1
virtualenv [-p python][--relocatable] DEST_DIR  # 2
source DEST_DIR/bin/activate                            # 3
deactivate
```

virtualenvwrapper

```
pip install virtualenvwrapper
export WORKON_HOME=~/Envs
source `which virtualenvwrapper.sh`

mkvirtualenv venv
workon venv
export PROJECT_HOME=~/Projects
mkproject myproject
deactivate
rmvirtualenv venv

cdsitepackages
lssitepackages
```

autoenv

When you `cd` into a directory containing a `.env`, autoenv automagically activates the environment.

```
pip install autoenv
echo "source `which activate.sh`" >> ~/.bashrc

$ touch project/.env
$ echo "echo 'woah'" > project/.env
$ cd project
woah
```

https://github.com/kennethreitz/autoenv
http://docs.python-guide.org/en/latest/dev/virtualenvs/

pip

```
pip list
pip freeze > requirements.txt        # 1
pip install -r requirements.txt      # 2
```

- ИСТОЧНИКИ
  - Python Package Index (pypi)
  - github, выбранную ветку или коммит
  - local, URL
- Python wheels
  - `http://pythonwheels.com/`
  - упрощает сборку пакетов, содержащих компилируемый C-код

anaconda, miniconda by Continuum analytics

`http://conda.pydata.org/miniconda.html`

- не только python (R, Cxx, language-agnostic)
- прекомпилированые пакеты под linux, osx, windows
- свой механизм поддержки виртуальных окружений
- включает pip
- комьюнити

Пример

```
git clone https://github.com/ContinuumIO/topik
cd topik
conda env create
source activate topik
conda list

\url{https://www.continuum.io/content/conda-data-science}
```

Окружения

- environment.yml

```
$ cat environment.yml
name: topik
channels:
  - memex
  - defaults
dependencies:
  - blaze=0.9.1=py27_0
  - elasticsearch
  ...
  - elasticsearch
  - pyldavis
```

Окружения-2

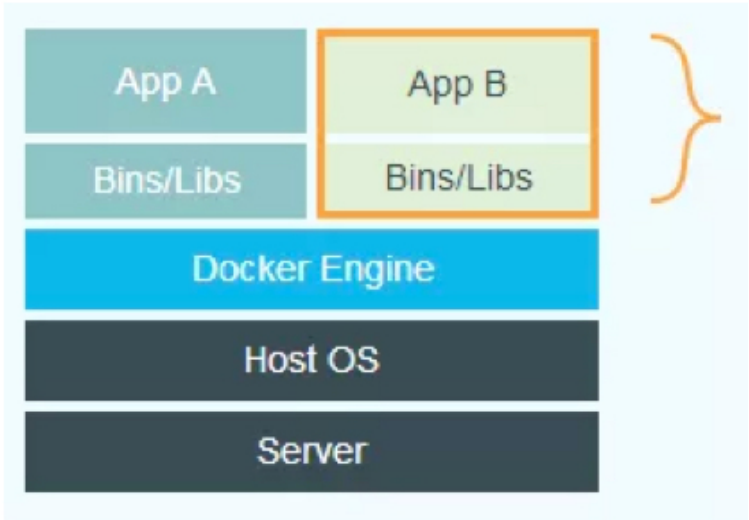
```
conda env list                                  # 1
conda create -n env_name python=3.5             # 2
conda env update -n env_name -f file.yml        # 3
source activate env_name                        # 4
source deactivate                               # 5

conda clean -a
```

Docker intro

> The kernel is a part of the operating system that handles communication with the hardware. It's the lowest level of the operating system. Here is a list of the main functions of the kernel:

- memory management
- network management
- device driver
- file management
- process management



Docker basics (Linux)

https://docs.docker.com/engine/installation/linux/ubuntulinux/

```
docker pull busybox
docker run -ti --rm busybox
docker ps -a
docker inspect CONTAINER_ID
docker exec -ti CONTAINER_ID COMMAND
docker version
```

https://docker.com/tryit/

Docker basics (OS X)

`https://kitematic.com/`, Windows, OS X

- Integration with VirtualBox
- Docker Hub Integration
- Seamless Experience Between CLI and GUI
- Advanced Features included

Docker under the hood (OS X)

- создает VM VirtualBox (2GB)
- IP: 192.168.99.100
- CLI
  - docker-machine start default
  - eval $(docker-machine env default)
  - docker-machine ssh default

Docker images

```
docker images
docker run -ti IMAGE
...
docker commit NEW_IMAGE
docker push NEW_IMAGE
docker rmi NEW_IMAGE
```

Docker containers

```
docker run -d IMAGE
docker ps
docker stop CONTAINER
docker rm CONTAINER
```

Port forwarding

```
docker run -d -p 8888:80 kitematic/hello-world-nginx
docker inspect CONTAINER

"Ports": {
    "80/tcp": [
        {
            "HostIp": "0.0.0.0",
            "HostPort": "8888"
        }
    ]
},

open http://192.168.99.100:8888
```

Disk sharing

```
echo echo "My index" > ~/website/index.html
docker run -d -p 8888:80 -v ~/website/:/website_files \
  kitematic/hello-world-nginx
docker inspect

"Mounts": [
    {
        "Source": "/Users/anaderi/website",
        "Destination": "/website_files",
        "Mode": "",
        "RW": true,
    }
],
```

Docker file system

- incremental layers
- Aufs (AnotherUnionFS), `https://ru.wikipedia.org/wiki/Aufs`
- content-addressable
- /var/lib/docker

Dockerfile

https://github.com/yandex/rep/blob/master/ci/Dockerfile.base
https://docs.docker.com/engine/reference/builder/

```
FROM ubuntu:14.04
MAINTAINER Andrey Ustyuzhanin <anaderi@yandex-team.ru>
WORKDIR /root
ADD ./* /root/
RUN /root/install-repbase-env.sh
RUN echo " \n\
  export PATH=$HOME/miniconda/bin:\$PATH \n\
  source activate rep_py2 \n\
  pushd \$ENV_BIN_DIR/.. ; source 'bin/thisroot.sh' ; pop
" >> /etc/profile.d/rep_profile.sh
CMD ["bash"]
```

Build image

```
docker build [-t TAG] PATH | URL | -
```

Dockerhub

- docker push
- autbuild (webhook)
- build triggers (dependencies)
- Local docker registry
  `https://docs.docker.com/registry/deploying/`

ReproZip

ReproZip allows you to pack your experiment along with all necessary data files, libraries, environment variables and options.
`https://vida-nyu.github.io/reprozip/`

```
pip install reprozip # Linux-only
reprozip trace ./myexperiment -my --options \
    inputs/somefile.csv other_file_here.bin
reprozip pack my_experiment.rpz

pip install reprounzip[all]
reprounzip docker setup my_experiment.rpz mydirectory
reprounzip docker run mydirectory
```

Спасибо за внимание!