

snakemake lecture

+rosalind 1:10:06 Lera Inferring peptide from full spectrum +Dima

common workflow language: explicit graph descr: dep (links) b/w operations
sm:implicit:dep b/w data

-graphviz to visualize pipeline

```
[root@octopus octopus] Oct 27 11:50 [omitted].out
(base) octopus@octopus-ThinkPad-T480s:~/Documents/2scripts/snakefile/my_additional_files$ snakemake -s big_pipeline --dag | dot -Tsvg > vizOfBigPipeline.svg
```

```
azat@azat-HP-Notebook:~$ snakemake -s big_pipeline --dag | dot -Tsvg > vizOfBigPipeline.svg
Error: Snakefile "big_pipeline" not present.
azat@azat-HP-Notebook:~$ vim somefile
azat@azat-HP-Notebook:~$ snakemake -s somefile
SyntaxError in line 2 of /home/azat/somefile:
    Invalid syntax
azat@azat-HP-Notebook:~$ vim somefile
azat@azat-HP-Notebook:~$ snakemake -s somefile
SyntaxError in line 4 of /home/azat/somefile:
    Invalid syntax
azat@azat-HP-Notebook:~$ vim somefile
azat@azat-HP-Notebook:~$ snakemake -s somefile
Provided cores: 1
Rules claiming more threads will be scaled down.
Job counts:
    count   jobs
        1     all
        1
rule all:
    output: out.txt
1 of 1 steps (100%) done
azat@azat-HP-Notebook:~$ snakemake -s somefile --dag | dot -Tsvg > vizNew.svg
Команда 'snakemake' не найдена, возможно вы имели в виду:
    Команда 'snakemake' из пакета 'snakemake' (universe)
snakemake: команда не найдена
azat@azat-HP-Notebook:~$ snakemake -s somefile --dag | dot -Tsvg > vizNew.svg
```

added relations

```
rule copy:
    input: "out.txt"
    output: "copy.txt"
    shell: "cp {input} {output}"

rule generate:
    output: "out.txt"
    shell: "touch {output}"
~
```

```
azat@azat-HP-Notebook:~$ vim somefile
azat@azat-HP-Notebook:~$ snakemake -s somefile
Provided cores: 1
Rules claiming more threads will be scaled down.
Job counts:
    count      jobs
        1        copy
        1
rule copy:
    input: out.txt
    output: copy.txt
1 of 1 steps (100%) done
azat@azat-HP-Notebook:~$
```

```
azat@azat-HP-Notebook:~$ ls c*.txt
copy.txt
```

```
azat@azat-HP-Notebook:~$ snakemake -s somefile --dag | dot -Tsvg > vizNew2.svg
azat@azat-HP-Notebook:~$
```

change the order of rules and rm out.txt —> 1:58

Stepik 1 2 3 4 5

<https://stepik.org/lesson/35914/step/1?unit=15065>

```
rule copy:
    input: "in/{file}"
    output: "out/{file}"
    shell: "cp {input} {output}"
```

—> error

```
RuleException in line 1 of /home/azat/somefile:
Could not resolve wildcards in rule copy:
file
azat@azat-HP-Notebook:~$
```

wildcards constr allowing to match input with ooutput files

```
rule all:
    input: ["out/1", "out/2", "out/3", "out/4", "out/5"]
    output: ".status"
    shell: "touch {output}"
rule copy:
    input: "in/{file}"
    output: "out/{file}"
    shell: "cp {input} {output}"
```

result

```

zat@azat-HP-Notebook:~$ snakemake -s somefile
Provided cores: 1
Rules claiming more threads will be scaled down.
Job counts:
    count   jobs
    1       all
    5       copy
    6

rule copy:
    input: in/3
    output: out/3
of 6 steps (17%) done
rule copy:
    input: in/5
    output: out/5
of 6 steps (33%) done
rule copy:
    input: in/1
    output: out/1
of 6 steps (50%) done
rule copy:
    input: in/4
    output: out/4
of 6 steps (67%) done
rule copy:
    input: in/2
    output: out/2
of 6 steps (83%) done
rule all:
    input: out/1, out/2, out/3, out/4, out/5
    output: .status
of 6 steps (100%) done
zat@azat-HP-Notebook:~$ ls out
2 3 4 5
zat@azat-HP-Notebook:~$ ls in
2 3 4 5
zat@azat-HP-Notebook:~$
```

adding python def gen <https://stepik.org/lesson/35914/step/4?unit=15065>

```

def gen(wildcards):
    return ["out/{}".format(i) for i in range(1,6)]


rule all:
    input: gen
    output: ".status"
    shell: "touch {output}"
rule copy:
    input: "in/{file}"
    output: "out/{file}"
    shell: "cp {input} {output}"
```

docker

-reproducible programs for different os

-alternative for VM

Parts of docker:

1. hub -actual server with all images and their descriptions
 2. image -set of rules how container should work (эквиз)
 3. container
-

Practice

to get rid of errors:

```
docker-machine rm default  
docker-machine create default --driver virtualbox  
docker-machine env default  
eval $(docker-machine env default)
```

```
docker-machine stop default
```

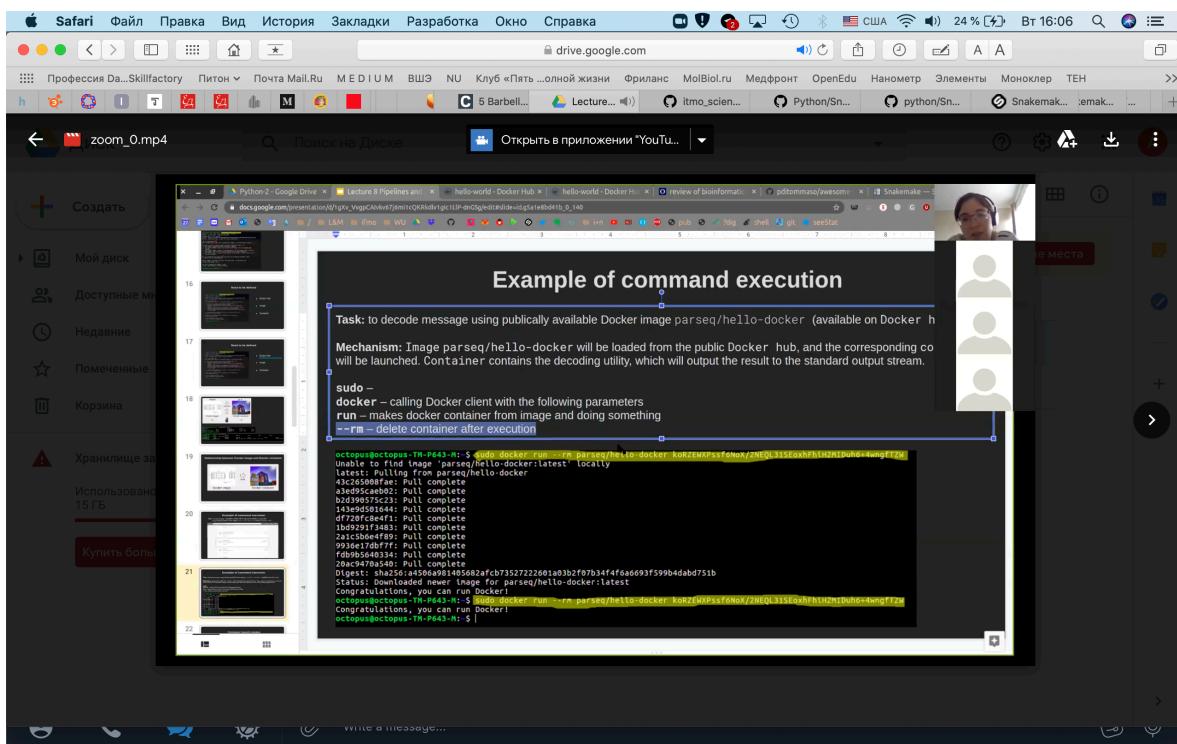
- installed Docker, Docker Machine, VirtualBox
 - docker-machine create created a new Docker virtual machine (named 'default')
 - docker-machine env switched your environment to your new VM (named 'default')
 - evaluse the docker client(your local computer/terminal) to create, load, and manage containers
 - docker run Docker client (your terminal) created the Docker deamon
 - pulled the first image 'hello-world'
 - The Docker deamon created a container, streamed out that output to the Docker client
 - docker-machine stop shut down the virtual machine
-

commands from lecture to try:

```
sudo docker hello-world  
sudo docker images
```

Example:

decode message using image parseq from hub



Interactive mode

`sudo docker —rm —it [interactive]...`

-Go from container library to home folder: exit

Structure of filesystem

-Layered

-Changes of files=new layer

-Copy-on-write rule: original file is kept in deeper layers

-`sudo docker history [name of container]`-list of layers

Keeping data outside the C. 50 min video

-we can mount folders to our container

`sudo docker —rm —it -v[mount] $<(dir1)>:<dir2>`

Task

You can mount not only directories, but also individual files, the syntax is not changed.

The program in the parseq/stepik-mount-files image writes something important in /dev/null. Try to mount an arbitrary host file instead of /dev/null when starting the container and [send a message](#) from the file as a response.

solution:

```
Lecture5_27.04.2020 – Google Диск
octopus@octopus-ThinkPad-T480s:~$ sudo docker run --rm -v /home/octopus/someFile:/dev/null parseq/stepik-mount-files
octopus@octopus-ThinkPad-T480s:~$ cat someFile
42
```

*superusers rights

Quiz

Send 5 T/F to @mini_oct

- | | |
|---|---|
| 1 | There may be an image for which there is no container. |
| 2 | If the image is not on the local machine, attempting to create a container will result in an error. |
| 3 | Running the application in a container allows you to protect the operating system from making changes. |
| 4 | Due to the use of copy-on-write, it is possible to change the file system layers included in the container image. |
| 5 | An implicit way of pipeline describing implies the absence of direct connections between stages. |