Ch-134.DevOps/Cisco

Demo 1

IT Academy

softserve

Team 1

Vladyslav Boreiko

Ivan Kuvila

Ch-134.DevOps/Cisco

softserve

Vladyslav Boreiko

GitHub

Python

Geo Citizen

DaC

softserve

Python

Modules:

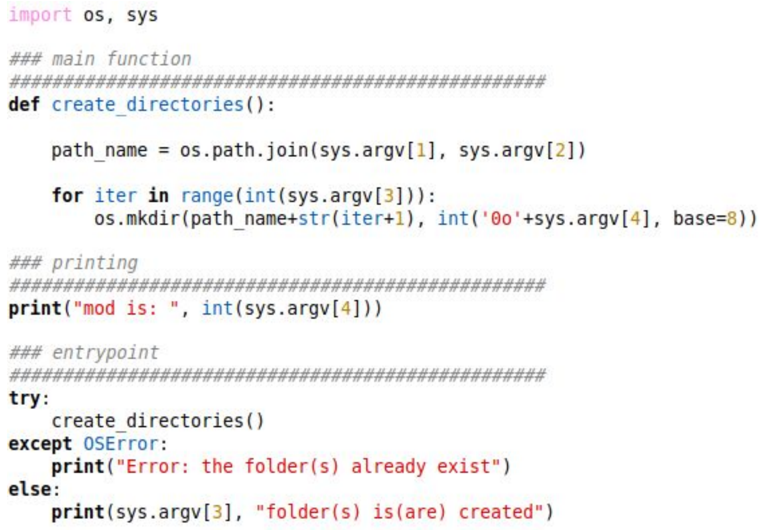
1. Operating System interfaces
2. Text
3. Networking
4. Packaging
5. Databases
6. Docker

softserve

Python 1: Operating System interfaces

Task:

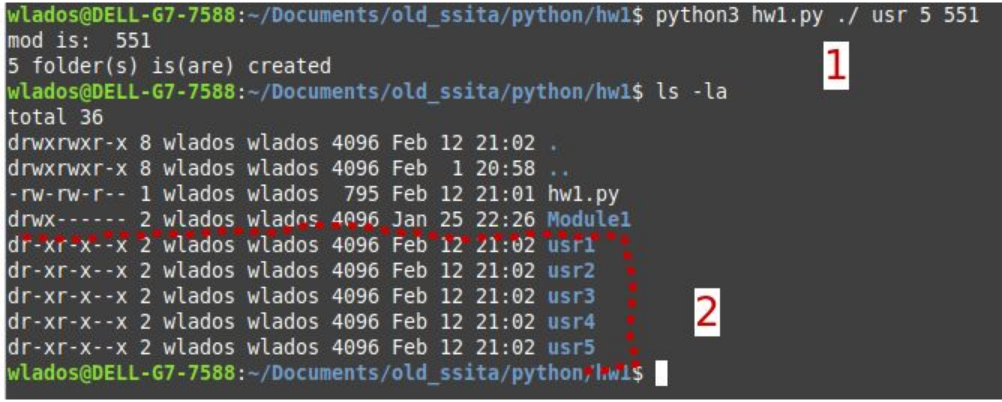
Create a program that generate *folders*.



```python
import os, sys

### main function
##################################################
def create_directories():

    path_name = os.path.join(sys.argv[1], sys.argv[2])

    for iter in range(int(sys.argv[3])):
        os.mkdir(path_name+str(iter+1), int('0o'+sys.argv[4], base=8))

### printing
##################################################
print("mod is: ", int(sys.argv[4]))

### entrypoint
##################################################
try:
    create_directories()
except OSError:
    print("Error: the folder(s) already exist")
else:
    print(sys.argv[3], "folder(s) is(are) created")
```

Result of example run:

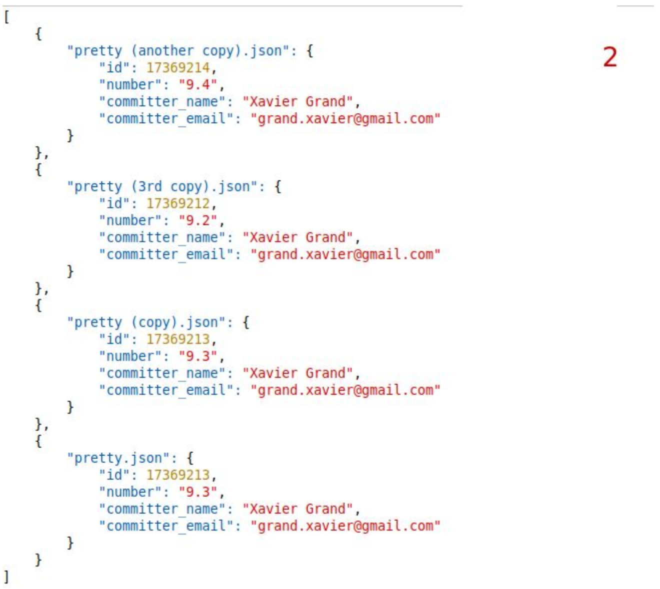
It creates 20 folders on the path /home with names usr1, usr2, etc. and permissions mode 551



```
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw1$ python3 hw1.py ./ usr 5 551
mod is:  551
5 folder(s) is(are) created
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw1$ ls -la
total 36
drwxrwxr-x 8 wlados wlados 4096 Feb 12 21:02 .
drwxrwxr-x 8 wlados wlados 4096 Feb  1 20:58 ..
-rw-rw-r-- 1 wlados wlados  795 Feb 12 21:01 hw1.py
drwx------ 2 wlados wlados 4096 Jan 25 22:26 Module1
dr-xr-x--x 2 wlados wlados 4096 Feb 12 21:02 usr1
dr-xr-x--x 2 wlados wlados 4096 Feb 12 21:02 usr2
dr-xr-x--x 2 wlados wlados 4096 Feb 12 21:02 usr3
dr-xr-x--x 2 wlados wlados 4096 Feb 12 21:02 usr4
dr-xr-x--x 2 wlados wlados 4096 Feb 12 21:02 usr5
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw1$
```

softserve

Python 2: Text



```
[
    {
        "pretty (another copy).json": {
            "id": 17369214,
            "number": "9.4",
            "committer_name": "Xavier Grand",
            "committer_email": "grand.xavier@gmail.com"
        }
    },
    {
        "pretty (3rd copy).json": {
            "id": 17369212,
            "number": "9.2",
            "committer_name": "Xavier Grand",
            "committer_email": "grand.xavier@gmail.com"
        }
    },
    {
        "pretty (copy).json": {
            "id": 17369213,
            "number": "9.3",
            "committer_name": "Xavier Grand",
            "committer_email": "grand.xavier@gmail.com"
        }
    },
    {
        "pretty.json": {
            "id": 17369213,
            "number": "9.3",
            "committer_name": "Xavier Grand",
            "committer_email": "grand.xavier@gmail.com"
        }
    }
]
```

2

Task:

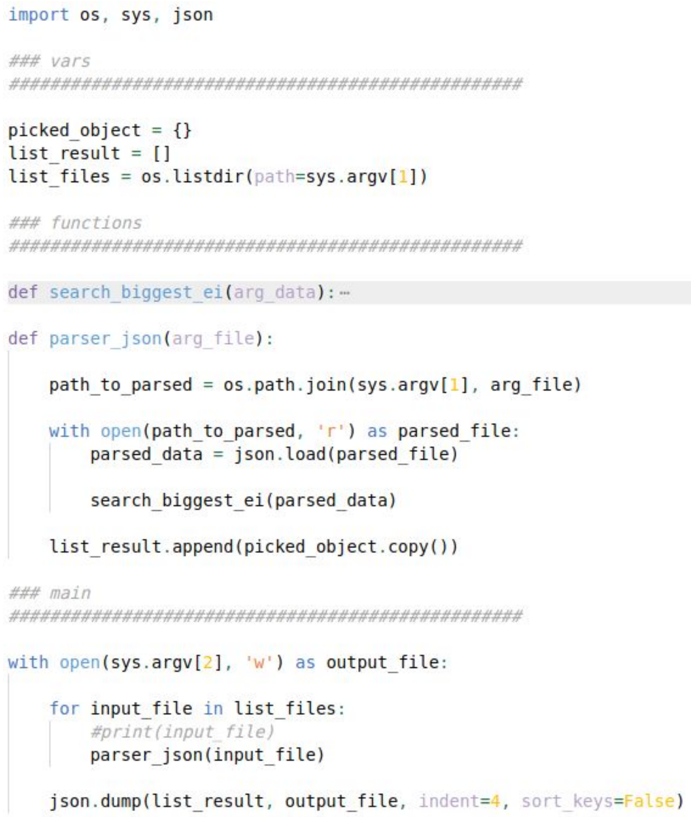
There are *a set of JSON-files* that contains answers from the CI server. An example of such is attached hw2_example.json. Create a program that returns JSON-file which contains:

- 'id',
- 'number',
- 'committer_name'
- 'committer_email'

from last of failed builds (in other words - with the highest value of 'number' and non-zero 'result').

Result of example run:

Result of example run: it reads all files on the path /home/usr/data_json and writes on the file /home/usr/result.json the necessary information.

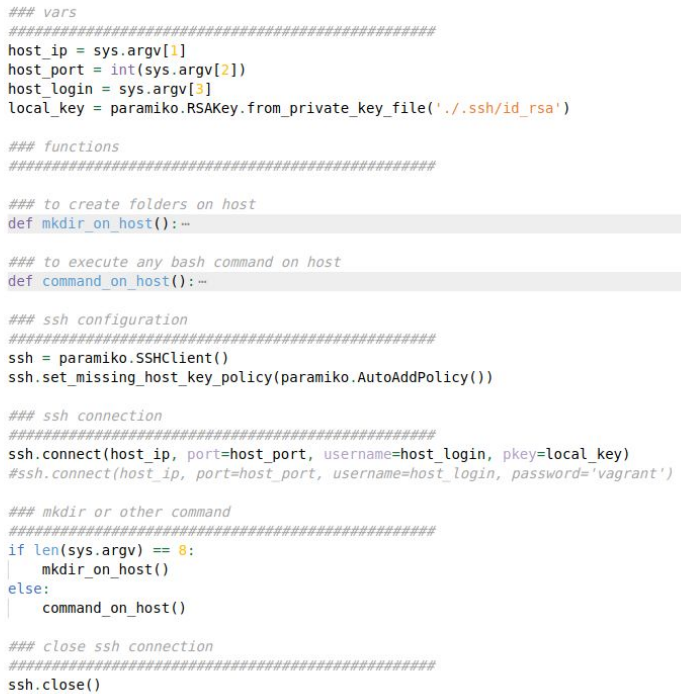


```
import os, sys, json

### vars
##################################################

picked_object = {}
list_result = []
list_files = os.listdir(path=sys.argv[1])

### functions
##################################################

def search_biggest_ei(arg_data):⋯

def parser_json(arg_file):

    path_to_parsed = os.path.join(sys.argv[1], arg_file)

    with open(path_to_parsed, 'r') as parsed_file:
        parsed_data = json.load(parsed_file)

        search_biggest_ei(parsed_data)

    list_result.append(picked_object.copy())

### main
##################################################

with open(sys.argv[2], 'w') as output_file:

    for input_file in list_files:
        #print(input_file)
        parser_json(input_file)

    json.dump(list_result, output_file, indent=4, sort_keys=False)
```

Vladyslav Boreiko

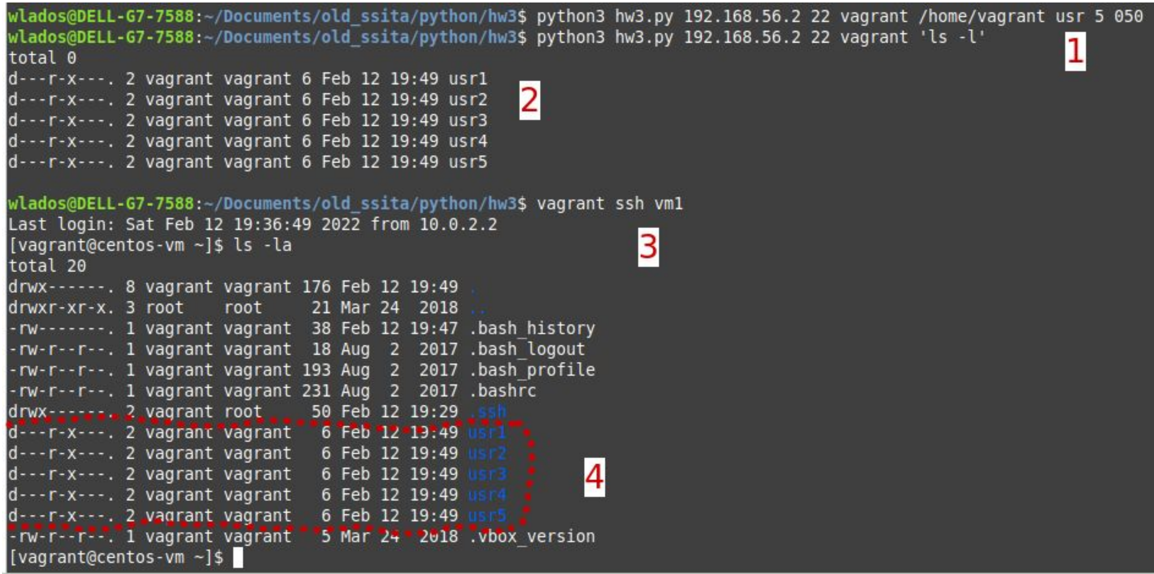
Python 3: Networking

Task:
Create a program that generate folders on *a remote computer* through a SSH connection.



```
### vars
##################################################
host_ip = sys.argv[1]
host_port = int(sys.argv[2])
host_login = sys.argv[3]
local_key = paramiko.RSAKey.from_private_key_file('./.ssh/id_rsa')

### functions
##################################################

### to create folders on host
def mkdir_on_host():…

### to execute any bash command on host
def command_on_host():…

### ssh configuration
##################################################
ssh = paramiko.SSHClient()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

### ssh connection
##################################################
ssh.connect(host_ip, port=host_port, username=host_login, pkey=local_key)
#ssh.connect(host_ip, port=host_port, username=host_login, password='vagrant')

### mkdir or other command
##################################################
if len(sys.argv) == 8:
    mkdir_on_host()
else:
    command_on_host()

### close ssh connection
##################################################
ssh.close()
```

Result of example run:
it runs ssh-connect to a remote host 192.168.0.2 using credential of 'someuser' and creates there: 20 folders, on the path /home, with names usr* and permissions mode 551.



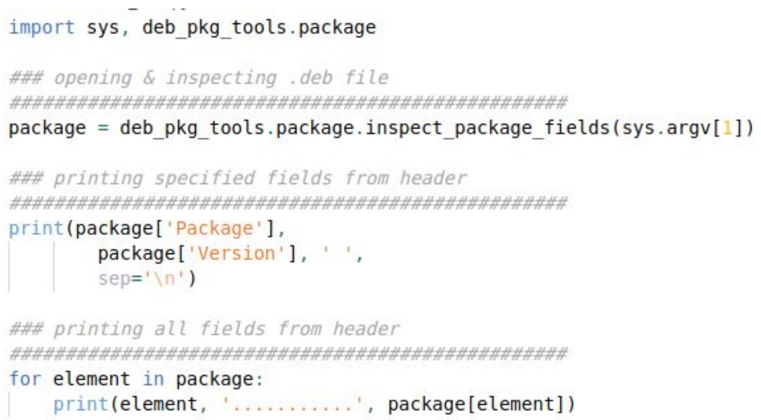
```
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw3$ python3 hw3.py 192.168.56.2 22 vagrant /home/vagrant usr 5 050
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw3$ python3 hw3.py 192.168.56.2 22 vagrant 'ls -l'
total 0
d---r-x---. 2 vagrant vagrant 6 Feb 12 19:49 usr1
d---r-x---. 2 vagrant vagrant 6 Feb 12 19:49 usr2
d---r-x---. 2 vagrant vagrant 6 Feb 12 19:49 usr3
d---r-x---. 2 vagrant vagrant 6 Feb 12 19:49 usr4
d---r-x---. 2 vagrant vagrant 6 Feb 12 19:49 usr5

wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw3$ vagrant ssh vm1
Last login: Sat Feb 12 19:36:49 2022 from 10.0.2.2
[vagrant@centos-vm ~]$ ls -la
total 20
drwx------. 8 vagrant vagrant 176 Feb 12 19:49 .
drwxr-xr-x. 3 root    root     21 Mar 24  2018 ..
-rw-------. 1 vagrant vagrant  38 Feb 12 19:47 .bash_history
-rw-r--r--. 1 vagrant vagrant  18 Aug  2  2017 .bash_logout
-rw-r--r--. 1 vagrant vagrant 193 Aug  2  2017 .bash_profile
-rw-r--r--. 1 vagrant vagrant 231 Aug  2  2017 .bashrc
drwx------. 2 vagrant root     50 Feb 12 19:29 .ssh
d---r-x---. 2 vagrant vagrant   6 Feb 12 19:49 usr1
d---r-x---. 2 vagrant vagrant   6 Feb 12 19:49 usr2
d---r-x---. 2 vagrant vagrant   6 Feb 12 19:49 usr3
d---r-x---. 2 vagrant vagrant   6 Feb 12 19:49 usr4
d---r-x---. 2 vagrant vagrant   6 Feb 12 19:49 usr5
-rw-r--r--. 1 vagrant vagrant   5 Mar 24  2018 .vbox_version
[vagrant@centos-vm ~]$
```

Vladyslav Boreiko

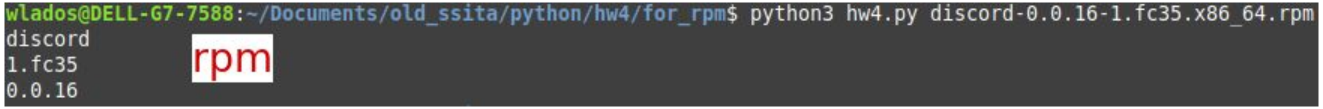
Python 4: Packaging

Task:
There is some *rpm-file*. Create program that outputs header field rpm.RPMTAG_RELEASE of this file.

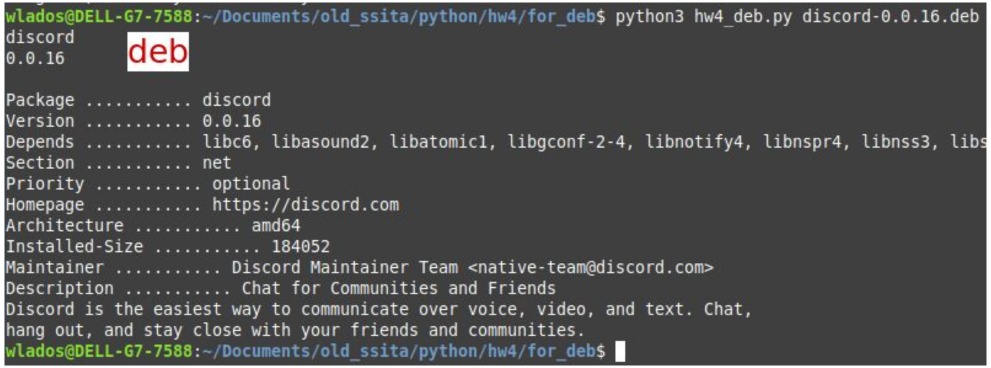
Result of example run:
it reads header of the file /home/usr/some_file.rpm and print field like this: 5.rel8.centos



```
import sys, deb_pkg_tools.package

### opening & inspecting .deb file
##################################################
package = deb_pkg_tools.package.inspect_package_fields(sys.argv[1])

### printing specified fields from header
##################################################
print(package['Package'],
        package['Version'], ' ',
        sep='\n')

### printing all fields from header
##################################################
for element in package:
    print(element, '...........', package[element])
```



```
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw4/for_rpm$ python3 hw4.py discord-0.0.16-1.fc35.x86_64.rpm
discord
1.fc35
0.0.16
```

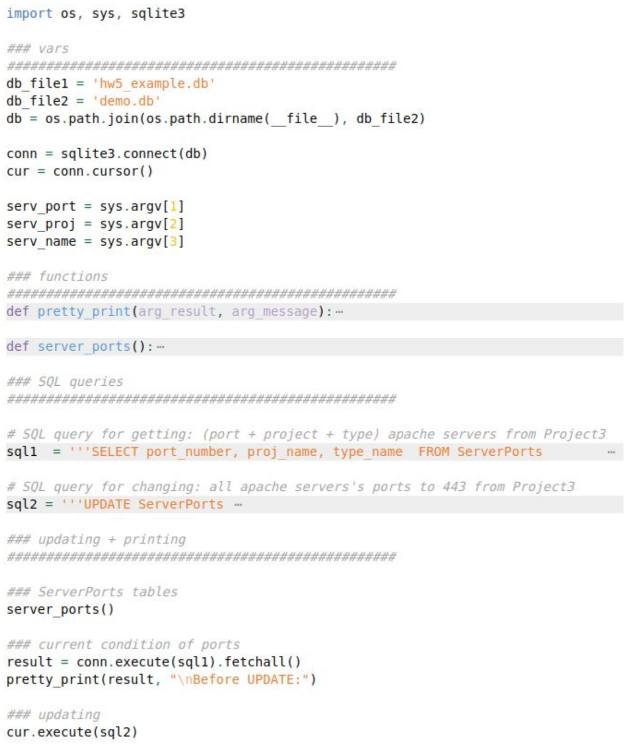


```
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw4/for_deb$ python3 hw4_deb.py discord-0.0.16.deb
discord
0.0.16

Package ........... discord
Version ........... 0.0.16
Depends ........... libc6, libasound2, libatomic1, libgconf-2-4, libnotify4, libnspr4, libnss3, libs
Section ........... net
Priority ........... optional
Homepage ........... https://discord.com
Architecture ........... amd64
Installed-Size ........... 184052
Maintainer ........... Discord Maintainer Team <native-team@discord.com>
Description ........... Chat for Communities and Friends
Discord is the easiest way to communicate over voice, video, and text. Chat,
hang out, and stay close with your friends and communities.
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw4/for_deb$
```

softserve

Vladyslav Boreiko

Python 5: Database

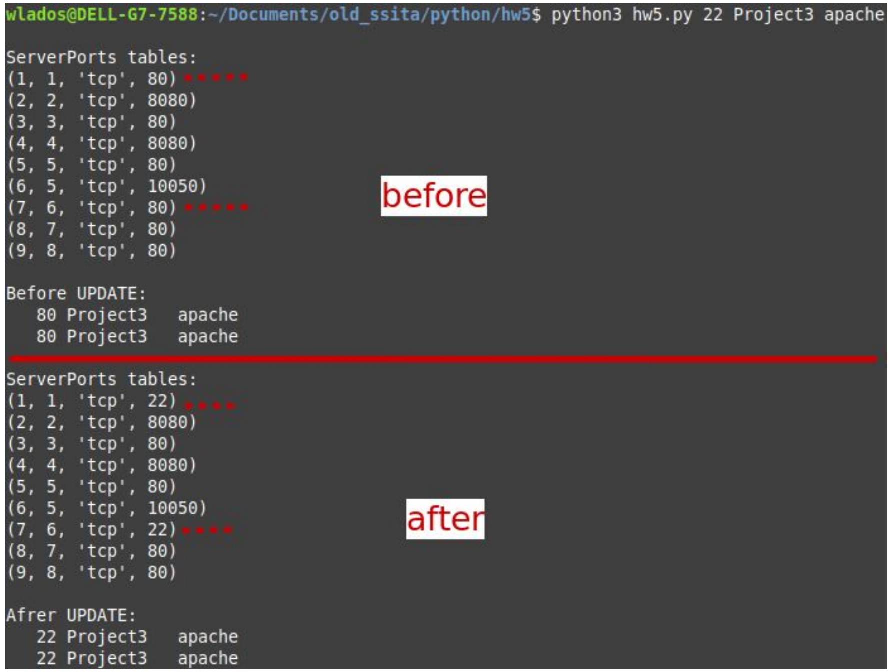


```
import os, sys, sqlite3

### vars
##################################################
db_file1 = 'hw5_example.db'
db_file2 = 'demo.db'
db = os.path.join(os.path.dirname(__file__), db_file2)

conn = sqlite3.connect(db)
cur = conn.cursor()

serv_port = sys.argv[1]
serv_proj = sys.argv[2]
serv_name = sys.argv[3]

### functions
##################################################
def pretty_print(arg_result, arg_message):…

def server_ports():…

### SQL queries
##################################################

# SQL query for getting: (port + project + type) apache servers from Project3
sql1  = '''SELECT port_number, proj_name, type_name  FROM ServerPorts         …

# SQL query for changing: all apache servers's ports to 443 from Project3
sql2 = '''UPDATE ServerPorts …

### updating + printing
##################################################

### ServerPorts tables
server_ports()

### current condition of ports
result = conn.execute(sql1).fetchall()
pretty_print(result, "\nBefore UPDATE:")

### updating
cur.execute(sql2)
```

Task:
There is some SQLite database *example.db*. Create program that sets:

- in database ports (ServerPorts.port_number)
- to 443
- for all servers apache (ServerTypes.type_name is 'apache')
- in project 'Project3'.

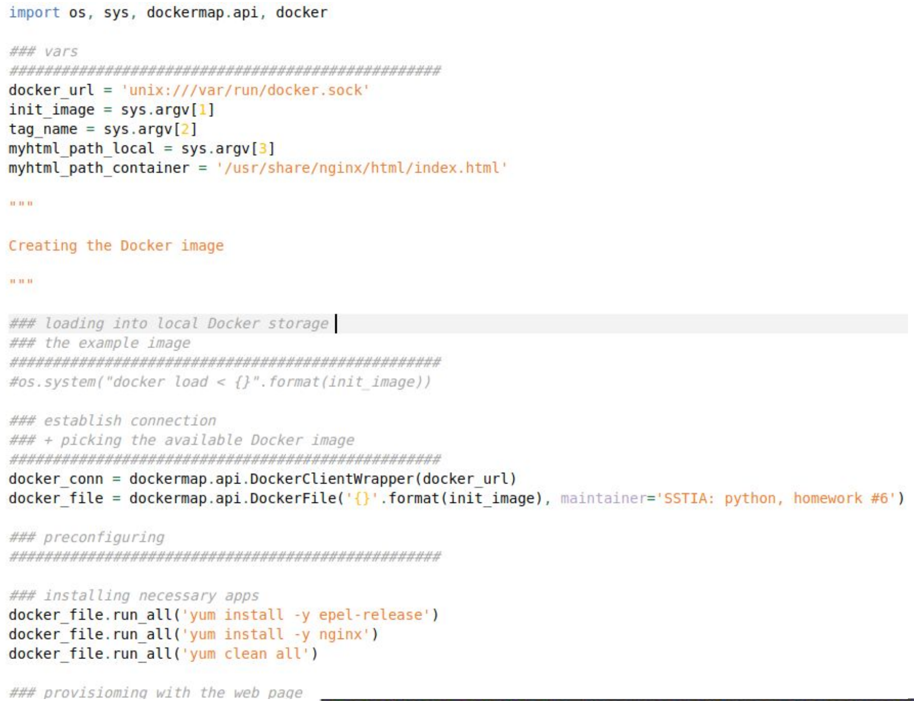
Result of example run:
It sets specified ports of some servers in certain project on input number.



```
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw5$ python3 hw5.py 22 Project3 apache

ServerPorts tables:
(1, 1, 'tcp', 80) *****
(2, 2, 'tcp', 8080)
(3, 3, 'tcp', 80)
(4, 4, 'tcp', 8080)
(5, 5, 'tcp', 80)
(6, 5, 'tcp', 10050)
(7, 6, 'tcp', 80) *****
(8, 7, 'tcp', 80)
(9, 8, 'tcp', 80)

Before UPDATE:
   80 Project3    apache
   80 Project3    apache

ServerPorts tables:
(1, 1, 'tcp', 22) ****
(2, 2, 'tcp', 8080)
(3, 3, 'tcp', 80)
(4, 4, 'tcp', 8080)
(5, 5, 'tcp', 80)
(6, 5, 'tcp', 10050)
(7, 6, 'tcp', 22) ****
(8, 7, 'tcp', 80)
(9, 8, 'tcp', 80)

Afrer UPDATE:
   22 Project3    apache
   22 Project3    apache
```

Vladyslav Boreiko

Python 6: Docker

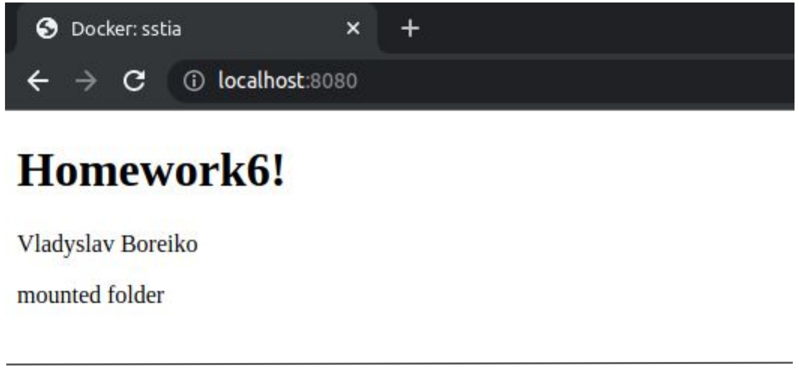


```
import os, sys, dockermap.api, docker

### vars
#####################################################
docker_url = 'unix:///var/run/docker.sock'
init_image = sys.argv[1]
tag_name = sys.argv[2]
myhtml_path_local = sys.argv[3]
myhtml_path_container = '/usr/share/nginx/html/index.html'

"""

Creating the Docker image

"""

### loading into local Docker storage
### the example image
#####################################################
#os.system("docker load < {}".format(init_image))

### establish connection
### + picking the available Docker image
#####################################################
docker_conn = dockermap.api.DockerClientWrapper(docker_url)
docker_file = dockermap.api.DockerFile('{}'.format(init_image), maintainer='SSTIA: python, homework #6')

### preconfiguring
#####################################################

### installing necessary apps
docker_file.run_all('yum install -y epel-release')
docker_file.run_all('yum install -y nginx')
docker_file.run_all('yum clean all')

### provisioming with the web page
```

Task:

Write program that *creates a Docker image* that based on image 'centos7/hw' (which need to import) and contains a simple Web applications is that displays in a browser "Homework6!".

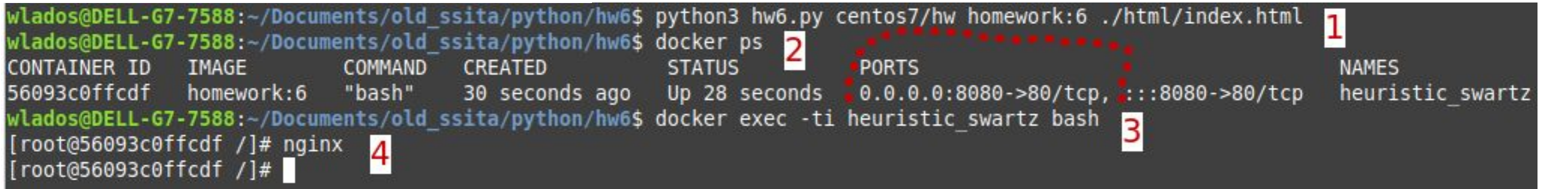
Result of example run:
it creates new Docker image with name 'homework:6' and based on image 'centos7/hw'. The command: 'docker run homework:6' starts the container. The connection to the address of one through the browser returns in browser string "Homework6!".



Homework6!

Vladyslav Boreiko

mounted folder



```
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw6$ python3 hw6.py centos7/hw homework:6 ./html/index.html  1
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw6$ docker ps  2
CONTAINER ID   IMAGE        COMMAND   CREATED          STATUS         PORTS                                  NAMES
56093c0ffcdf   homework:6   "bash"    30 seconds ago   Up 28 seconds  0.0.0.0:8080->80/tcp, :::8080->80/tcp   heuristic_swartz
wlados@DELL-G7-7588:~/Documents/old_ssita/python/hw6$ docker exec -ti heuristic_swartz bash  3
[root@56093c0ffcdf /]# nginx  4
[root@56093c0ffcdf /]# 
```

Vladyslav Boreiko

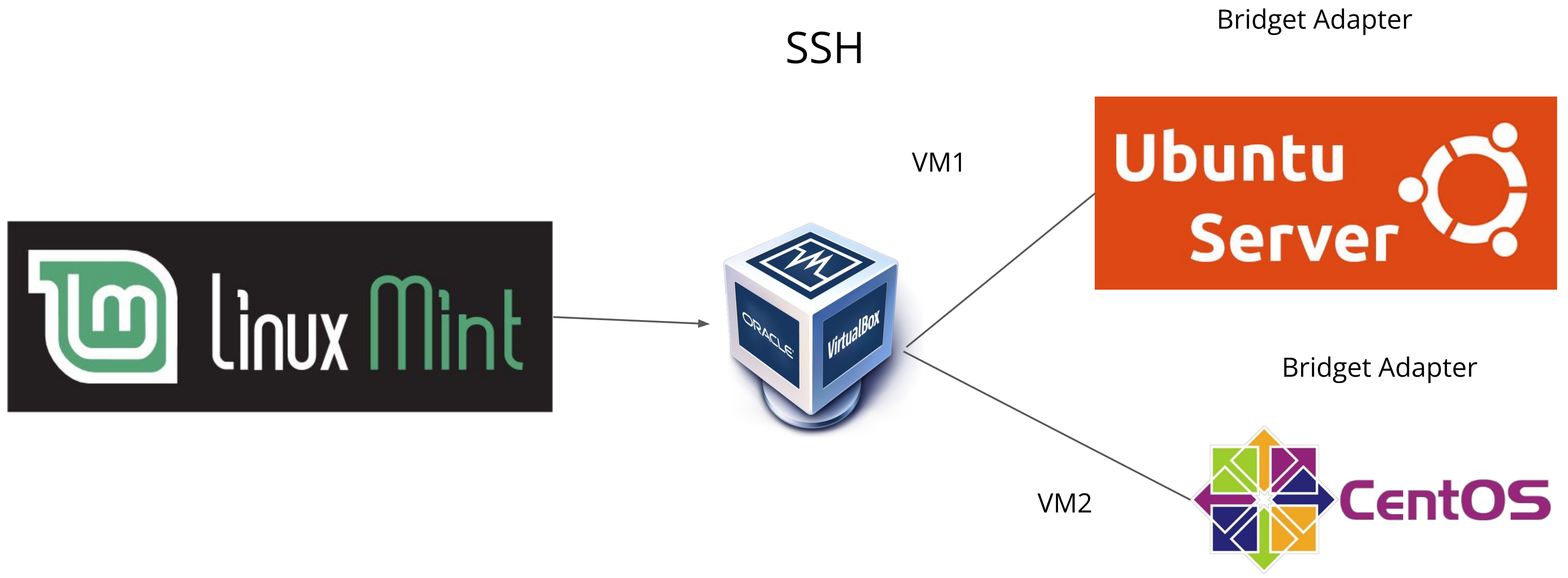
Geo Citizen

Prepare	Ubuntu - Server	CentOS - DB	The application (Geocit134)	Bash scripts
VMs	Openssh	Openssh	GitHub	To automate deploying
IP	Git	PostgreSQL	Fixing	by Bash
SSH	Java	geo-DB	Configs	as much
	Maven		Build	as possible
	Tomcat		Deploy	
	Geocit134			

softserve

Vladyslav Boreiko

Prepare



softserve

Vladyslav Boreiko

Ubuntu - Server

Ubuntu 20.04.3 Focal Fossa:

- OpenSSH_8.2p1
- openjdk 11.0.13 2021-10-19
- git 2.25.1
- Apache Tomcat 9.0.58
- Apache Maven 3.8.4

- Geocit134

CentOS - DB

CentOS 7.9.2009 Minimal:

- OpenSSH_7.4p1
- PostgreSQL 9.2.24

- User of Geocit134 DB
- DB of Geocit134
- Access for the user

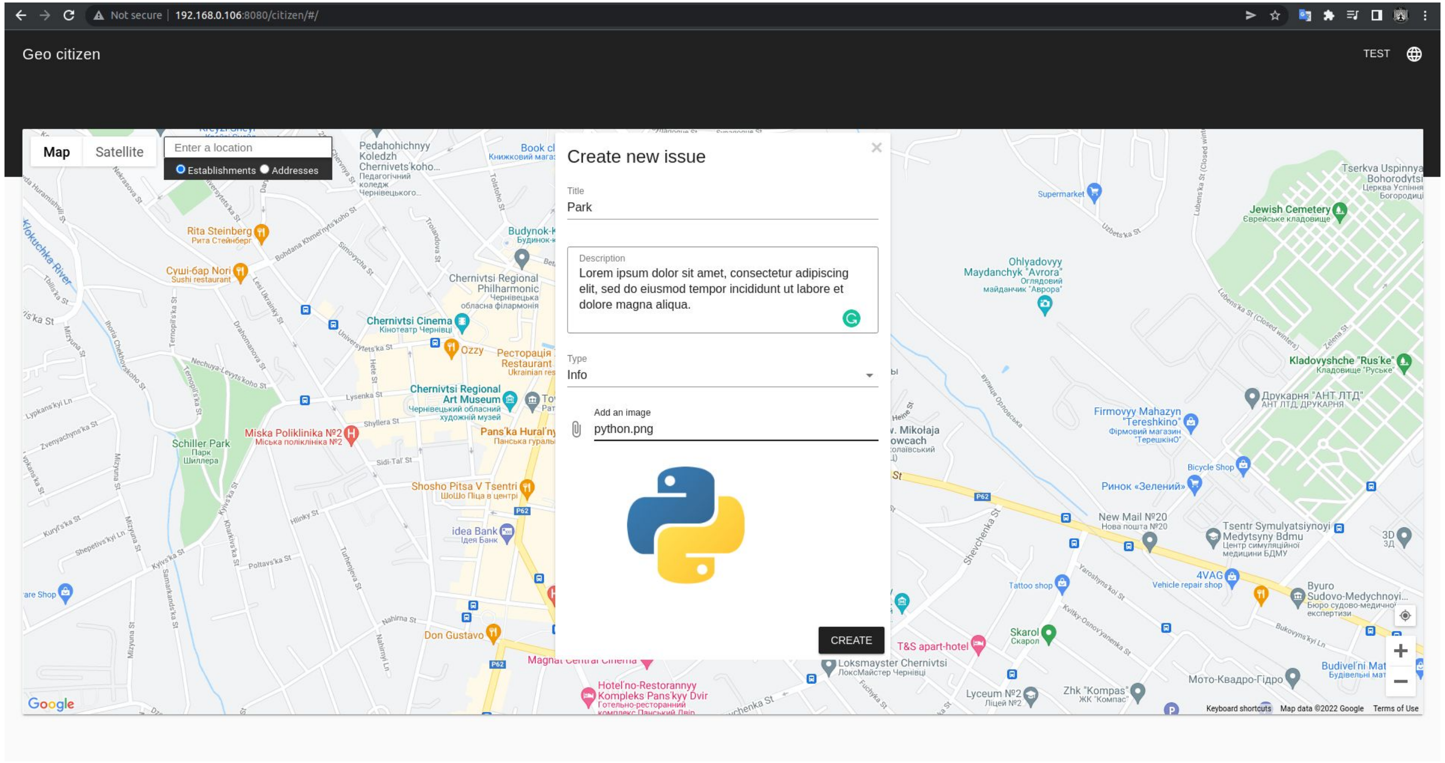
Vladyslav Boreiko

The project - Geocit134

Geocit134 on GitHub:

- Get by git
- First building -> many errors/warnings
- Fix
  - Paths
  - Duplicates
  - Plugin descriptions
  - Versions
  - Properties (hosts, credentials etc.)
  - Front-end (hosts, paths)
- Final building
- Deploy to Tomcat

The application - Geo Citizen



192.168.0.106:8080/citizen/#/

Geo citizen

Create new issue

Title
Park

Description
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Type
Info

Add an image
python.png

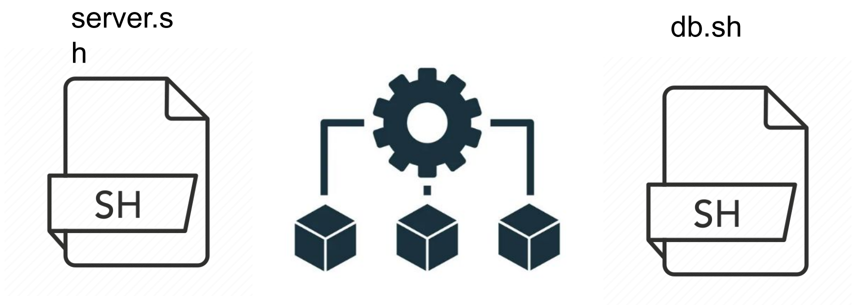
softserve

Vladyslav Boreiko

Bash scripts

Partial automatization of application deploying:

- server.sh
  - Removing the old project
  - Cloning the project again
  - Small errors fixing
  - Duplicates removing
  - Front-end fixing
  - The project building
  - The project deploying
- db.sh
  - Drop database
  - Drop role
  - Create role
  - Alter role
  - Create database
  - Grant access



softserve

Vladyslav Boreiko

Results

Objects:

- Working application
- 2 VMs
- SSH access to VMs

Recordings:

- Runbook Geocitizen.md
- Runbook VMs.md
- Jira issue
- GitHub repo

First automating:

- project.sh
- db.sh

softserve

DaC

- Python
- Mingrammer
- Graphviz

softserve

Task

A diagram have to illustrate:

1. Terraform - creates 2 hosts
2. Ansible - configuring hosts (DB and Geo app)
3. Jenkins - runner for Terraform and Ansible

Jenkins

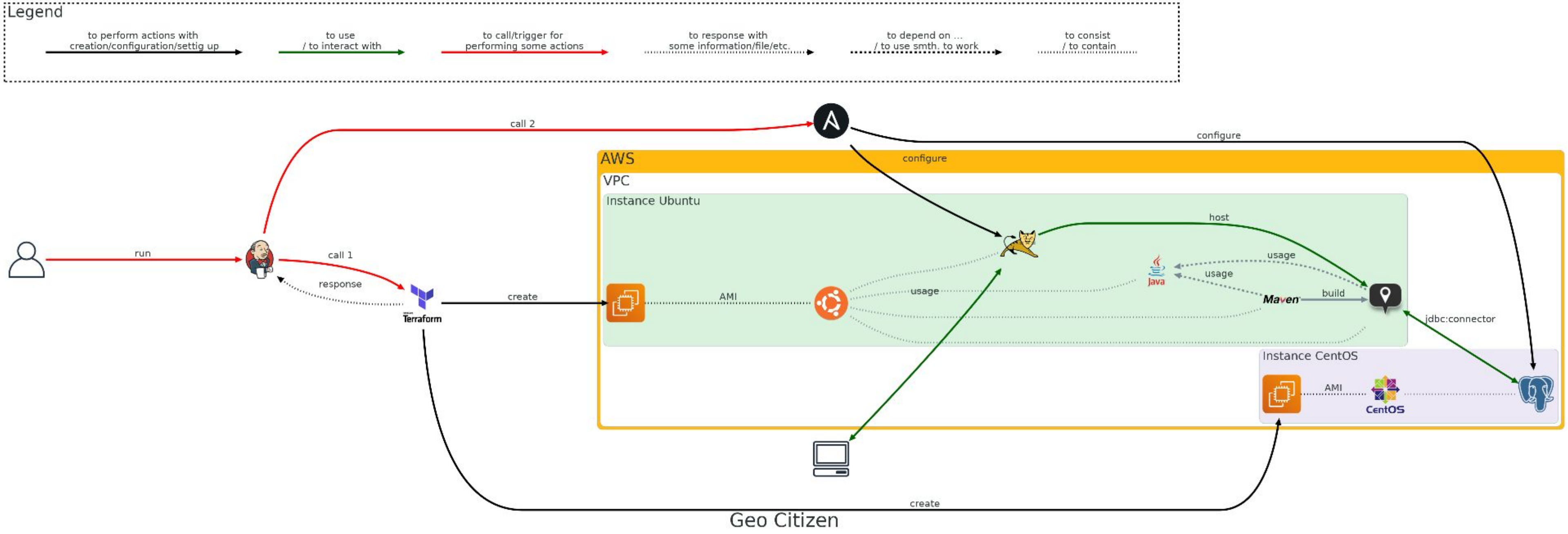
HashiCorp Terraform

ANSIBLE

PostgreSQL

softserve

Geocitizen - general diagram


Legend
to perform actions with creation/configuration/settig up
to use / to interact with
to call/trigger for performing some actions
to response with some information/file/etc.
to depend on ... / to use smth. to work
to consist / to contain
call 2
configure
configure
AWS
VPC
Instance Ubuntu
run
call 1
response
Terraform
create
AMI
usage
host
usage
usage
Java
Maven
build
jdbc:connector
Instance CentOS
AMI
CentOS
create
Geo Citizen

softserve

Vladyslav Boreiko

Any questions?