

Ivan Danyliuk

Python for DevOps



Git / GitHub



Manual Deploy Geocitizen Application



Python for DevOps

**introduction
to Python's capabilities
for system administration
and DevOps**

Module 1. Operating System interfaces

Module 2. Text

Module 3. Networking

Module 4. Packaging

Module 5. Databases

Ivan Danyliuk

Module 1. Operating System interfaces

Task: Create a program that generate folders.

Program code:

```
1 import os
2 import sys
3
4 path = sys.argv[1]
5 prefix = sys.argv[2]
6 counts = int(sys.argv[3])
7 mode = int("0o"+sys.argv[4], 8)
8
9 for n in range(1, counts+1):
10     dest = os.path.join(path, prefix + str(n))
11     try:
12         os.mkdir(dest, mode)
13     except OSError:
14         print('Oops, error!')
15     else:
16         print('Folder ' + dest + ' is created')
17
```

Result of example run: It creates 6 folders on the path /home/PyTest with names usr1, usr2, etc. and permissions mode 551

```
ivan@Dell-NB:~/PyTest$ /usr/bin/python3 /media/ivan/SYS/DevOPS/Python
/Module1/hw1.py ~/PyTest usr 6 551
Folder /home/ivan/PyTest/usr1 is created
Folder /home/ivan/PyTest/usr2 is created
Folder /home/ivan/PyTest/usr3 is created
Folder /home/ivan/PyTest/usr4 is created
Folder /home/ivan/PyTest/usr5 is created
Folder /home/ivan/PyTest/usr6 is created
ivan@Dell-NB:~/PyTest$ ls -al
загалом 36
drwxrwxr-x  9 ivan ivan 4096 лют 14 01:10 .
drwxr-xr-x 48 ivan ivan 4096 лют 13 23:47 ..
drwx-----  2 ivan ivan 4096 січ 25 23:44 releases
dr-xr-x--x  2 ivan ivan 4096 лют 14 01:10 usr1
dr-xr-x--x  2 ivan ivan 4096 лют 14 01:10 usr2
dr-xr-x--x  2 ivan ivan 4096 лют 14 01:10 usr3
dr-xr-x--x  2 ivan ivan 4096 лют 14 01:10 usr4
dr-xr-x--x  2 ivan ivan 4096 лют 14 01:10 usr5
dr-xr-x--x  2 ivan ivan 4096 лют 14 01:10 usr6
ivan@Dell-NB:~/PyTest$
```

Module 2. Text

Task: There are a set of JSON-files that contains answers from the CI server. Create a program that returns JSON-file which contains 'id', 'number', 'committer_name' and 'committer_email' from last of failed builds (in other words - with the highest value of 'number' and non-zero 'result').

Used module: json

Result of example run: it reads all files on the path entered in command prompt and writes on the result file (name also entered in command prompt) the necessary information like this:

```
{"id": 22, "number": "34", "committer_name": "Some Commiter",  
"committer_email": "some.commiter@gmail.com"}
```



The screenshot shows a code editor with a file named `hw2.py` containing a single line of JSON data: `{"id": 17369212, "number": "44", "committer_name": "Xavier Grand 2", "committer_email": "grand.xavier@gmail.com"}`. Below the editor is a terminal window with the command `/usr/bin/python3 /media/ivan/SYS/DevOPS/Python/hw/hw2.py ./files res.json` and its output.

```
hw2.py  {} res.json  X  
1 {"id": 17369212, "number": "44", "committer_name": "Xavier Grand 2", "committer_email": "grand.xavier@gmail.com"}  
  
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
ivan@Dell-NB: /media/ivan/SYS/DevOPS/Python/hw$ /usr/bin/python3 /media/ivan/SYS/DevOPS/Python/hw/hw2.py ./files res.json  
ivan@Dell-NB: /media/ivan/SYS/DevOPS/Python/hw$
```

Module 3. Networking

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

Problems:

1. SSH outputs result in binary format, so “ls -al” answer looks like

```
b'total 28\ndrwxrwxr-x  7 test test 4096 \xd0\xbb\xd1\x8e\xd1\x82 14 01:58 .\ndrwxr-xr-x 25 test test 4096 \xd0\xbb\xd1\x8e\xd1\x82 14 01:51 ..\ndrwxrw-r-x  2 test test 4096 \xd0\xbb\xd1\x8e\xd1\x82 14 01:58 dir1\ndrwxrw-r-x  2 test test 4096 \xd0\xbb\xd1\x8e\xd1\x82 14 01:58 dir2\ndrwxrw-r-x  2 test test 4096 \xd0\xbb\xd1\x8e\xd1\x82 14 01:58 dir3\ndrwxrw-r-x  2 test test 4096 \xd0\xbb\xd1\x8e\xd1\x82 14 01:58 dir4\ndrwxrw-r-x  2 test test 4096 \xd0\xbb\xd1\x8e\xd1\x82 14 01:58 dir5\n'
```

Method `.decode('utf-8')` was used for conversion.

2. Authentification used - login / password

Result of example run: It creates 5 folders on the path `~/PyTest` on remote host `10.1.1.108` via SSH connection with names `dir1`, `dir2`, etc. and permissions mode `765`

```
ivan@Dell-NB:~/media/ivan/SYS/DevOPS/Python/hw$ /usr/bin/python3 /media/ivan/SYS/DevOPS/Python/hw/hw3.py 10.1.1.108 22 "~/PyTest" dir 5 765
Enter user "test" password:
Password:

Directory ~/PyTest/dir1 successfully created ...

Directory ~/PyTest/dir2 successfully created ...

Directory ~/PyTest/dir3 successfully created ...

Directory ~/PyTest/dir4 successfully created ...

Directory ~/PyTest/dir5 successfully created ...
ivan@Dell-NB:~/media/ivan/SYS/DevOPS/Python/hw$
```

```
test@test:~$ ls -al ~/PyTest
total 28
drwxrwxr-x  7 test test 4096 лкТ 14 01:59 .
drwxr-xr-x 25 test test 4096 лкТ 14 01:51 ..
drwxrw-r-x  2 test test 4096 лкТ 14 01:59 dir1
drwxrw-r-x  2 test test 4096 лкТ 14 01:59 dir2
drwxrw-r-x  2 test test 4096 лкТ 14 01:59 dir3
drwxrw-r-x  2 test test 4096 лкТ 14 01:59 dir4
drwxrw-r-x  2 test test 4096 лкТ 14 01:59 dir5
```

Ivan Danyliuk

Module 3. Networking

```
1 import sys
2 import paramiko
3 import getpass
4
5 host_name = sys.argv[1]
6 port_name = int(sys.argv[2])
7 path = sys.argv[3]
8 prefix = sys.argv[4]
9 counts = int(sys.argv[5])
10 mode = sys.argv[6]
11
12 print('Enter username:')
13 user_name = input()
14 print('Enter user \'' + user_name + '\' password:')
15 user_password = getpass.getpass()
16 |
17 # start ssh connection
18 ssh = paramiko.SSHClient()
19 ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
20 ssh.connect(host_name, port_name, user_name, user_password,
21 |.....|.....look_for_keys=False, allow_agent=False)
22
23 # # test command 'ls -al' remotely and received output format
24 # stdin, stdout, stderr = ssh.exec_command('ls -al')
25 # res = stdout.read()
26 # print(type(res))
27 # print(res)
28 # print(type(str(res)))
29 # print(str(res))
30 # print(res.decode('utf-8'))
```

```
31
32
33 # # remove previously created 'path/prefix*' directories
34 stdin, stdout, stderr = ssh.exec_command('rmdir \'' + path + '/' + prefix + '*')
35 res = stdout.read().decode('utf-8') + stderr.read().decode('utf-8')
36 if res == '':
37 |.....|.....print('\nOld directories \'' + path + '/' +
38 |.....|.....prefix + '* successfully removed...')
39
40 # create required directories
41 for n in range(1, counts+1):
42 |.....|.....stdin, stdout, stderr = ssh.exec_command(
43 |.....|.....'mkdir -m \'' + mode + ' \'' + path + '/' + prefix + str(n))
44 |.....|.....res = stdout.read().decode('utf-8') + stderr.read().decode('utf-8')
45 |.....|.....if res == '':
46 |.....|.....|.....print('\nDirectory \'' + path + '/' + prefix +
47 |.....|.....|.....str(n) + ' successfully created...')
48 |.....|.....else:
49 |.....|.....|.....print(res)
50
51 # close ssh connection
52 ssh = ssh.invoke_shell()
53 ssh.close()
```

Module 4. Packaging

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

Problem: rpm module checks signatures, without keys we have no access.

```
ts = rpm.TransactionSet()
ts.setVSFlags(rpm._RPMVSF_NOSIGNATURES)
```

Result of example run: Info about rpm files in current directory

```
ivan@Dell-NB:/media/ivan/SYS/DevOPS/Python/hw$ /usr/bin/python3 /media/ivan/SYS/DevOPS/Python/hw/hw4.py

RPM package 'google-chrome-stable_current_x86_64.rpm' info
NAME.....google-chrome-stable
SUMMARY.....Google Chrome
VERSION.....97.0.4692.99
RELEASE.....1
ARCH.....x86_64
LICENSE.....Multiple, see https://chrome.google.com/
GROUP.....Applications/Internet
URL.....https://chrome.google.com/
PACKAGER.....Chrome Linux Team <chromium-dev@chromium.org>

RPM package 'some_package.el6.x86_64.rpm' info
NAME.....libreoffice-base
SUMMARY.....Database front-end for LibreOffice
VERSION.....4.0.4.2
RELEASE.....9.el6
ARCH.....x86_64
LICENSE.....(MPLv1.1 or LGPLv3+) and LGPLv3 and LGPLv2+ and BSD and (MPLv1.1 or GPLv2 or LGPLv2 or Netscape) and Public Domain and ASL 2.0 and Artistic and MPLv2.0
GROUP.....Applications/Productivity
URL.....http://www.libreoffice.org/default/
PACKAGER.....CentOS BuildSystem <http://bugs.centos.org>
```

Module 5. Databases

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'.

Program code:

Problems: 1. large SQL-query

```
sql_select_server = '''SELECT Servers.id, Projects.proj_name, Servers.dns_name,
Servers.ip_address, ServerTypes.type_name
FROM (((ServerProjects
INNER JOIN Projects ON Projects.id=ServerProjects.projects_id)
INNER JOIN Servers ON ServerProjects.servers_id=Servers.id)
INNER JOIN ServerTypes ON Servers.srvtypes_id=ServerTypes.id)
WHERE (Projects.proj_name=?) AND (ServerTypes.type_name=?);'''
```

2. Use .commit() to save data do DB.

Result of example run:

```
ivan@Dell-NB: /media/ivan/SYS/DevOPS/Python/hw$
/usr/bin/python3 /media/ivan/SYS/DevOPS/Python/
hw/5/hw5.py
-----
Selected servers:
-----
Server id: 1
Project name: Project3
Server DNS name: apache.my.local
Server IP address: 192.168.68.28
Server type: apache
-----
Server id: 6
Project name: Project3
Server DNS name: apache1.my.local
Server IP address: 192.168.68.30
Server type: apache
-----
Adding new port 443 to selected servers...
Server with id = 1 successfully added.
Server with id = 6 successfully added.
```

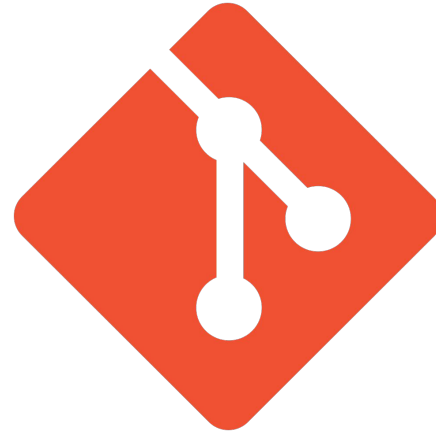

Module 5. Databases

```
1 import os
2 import sqlite3
3
4 sql_select_server = '''SELECT Servers.id, Projects.proj_name, Servers.dns_name,
5     Servers.ip_address, ServerTypes.type_name
6     FROM (((ServerProjects
7     INNER JOIN Projects ON Projects.id=ServerProjects.projects_id)
8     INNER JOIN Servers ON ServerProjects.servers_id=Servers.id)
9     INNER JOIN ServerTypes ON Servers.servertypes_id=ServerTypes.id)
10    WHERE (Projects.proj_name=?) AND (ServerTypes.type_name=?);'''
11 sql_insert_ports = '''INSERT INTO ServerPorts (servers_id, port_type, port_num)
12 sql_select_server_ports = '''SELECT ServerPorts.id FROM ServerPorts WHERE (serv
13
14
15 def check_server_ports(conn, params):
16     result = conn.execute(sql_select_server_ports, params).fetchall()
17     return len(result)
18
19
20 try:
21     db = os.path.join(os.path.dirname(__file__), 'demo.db')
22     conn = sqlite3.connect(db)
23 except:
24     print('Can\'t connect to Database ...')
25
26
27 selected_servers = conn.execute(
28     sql_select_server, ('Project3', 'apache')).fetchall()
29
30 print('-----')
31 print('Selected servers:')
```

```
33 for row in selected_servers:
34     print('-----')
35     print('Server id: ' + str(row[0]))
36     print('Project name: ' + row[1])
37     print('Server DNS name: ' + row[2])
38     print('Server IP address: ' + row[3])
39     print('Server type: ' + row[4])
40
41 print('-----')
42 print('Adding new port 443 to selected servers...')
43 for row in selected_servers:
44     if (check_server_ports(conn, (row[0], 443)) != 0):
45         print('Server with id = ' +
46             str(row[0]) + ' is already in database. Nothing to add ...')
47     else:
48         try:
49             conn.execute(sql_insert_ports, (row[0], 'tcp', 443))
50             conn.commit()
51             print('Server with id = ' + str(row[0]) + ' successfully added.')
52         except:
53             print('Error adding data to database ...')
54
55 conn.close()
```

Git / GitHub

Git was originally authored by *Linus Torvalds* in 2005 for development of the Linux kernel, with other kernel developers contributing to its initial development.

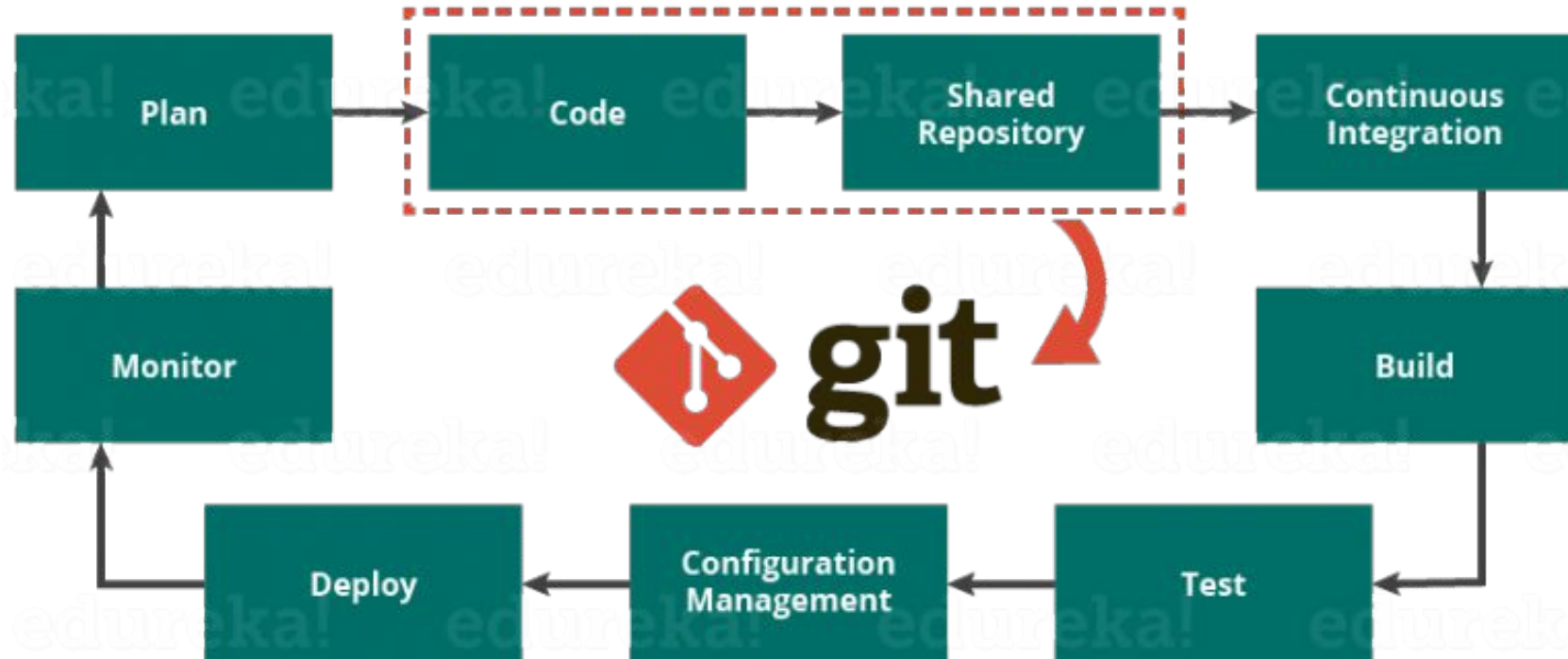


Git is just a version control system that manages and tracks changes to your source code

GitHub is a cloud-based hosting platform that manages all your Git repositories.

Role Of Git In DevOps

DevOps promotes communication between development engineers and operations, participating together in the entire service life-cycle, from design through the development process to production support.



Ivan Danyliuk

GitHub interaction variants

```
$ git clone https://github.com/<github_user>/<repository>.git
```

```
$ git config user.name <github_username>
```

```
$ git config user.email <github_username_email>
```

Save login/password locally

```
$ git config credential.helper store
```

If you have more than one GitHub accounts connected

```
$ git config credential.username <github_username>
```

Create new SSH key, using the provided GitHub email as a label. Add SSH key to the ssh-agent and to your account on GitHub.

```
$ git clone git@github.com:<github_user>/<repository>.git
```

Once you have selected a suitable distribution of Git for your platform, you'll need to identify yourself with a username and email address to Git. Use `--global` key to configure Git globally.

```
$ git config user.name <github_username>
```

```
$ git config user.email <github_username_email>
```

Ivan Danyliuk

VS Code with Git Graph extension

The screenshot shows the VS Code interface with the Git Graph extension. The Explorer sidebar on the left shows a project structure with folders like BashRegex, Diagrams, and Geocitizen. The main area displays the Git Graph view, which includes a commit history table and a visual graph of the commit history.

Branches: Show All ☒ Show Remote Branches

Graph	Description	Date	Author	Commit
main origin origin/HEAD	Merge pull request #13 from idanylyuk/DEV-...	14 Feb 2022 1...	idanylyuk	42e46a6e
DEV-76-data-visualization origin	add Diagrams	14 Feb 2022 1...	idanylyuk	353a1698
	Update README.md	14 Feb 2022 1...	idanylyuk	cca207f3
	Merge pull request #12 from idanylyuk/DEV-69-manual-deploy-geocitizen	13 Feb 2022 2...	idanylyuk	1fc1ef03
DEV-69-manual-deploy-geocitizen origin	Geo	13 Feb 2022 2...	idanylyuk	50da08e5
	Merge pull request #11 from idanylyuk/DEV-60-service-operation-basics	13 Feb 2022 2...	idanylyuk	fe04c724
DEV-60-service-operation-basics origin	BashRegex	13 Feb 2022 2...	idanylyuk	05d0d621
	rename dir	13 Feb 2022 2...	idanylyuk	a8ac9d48
	Merge pull request #10 from idanylyuk/DEV-50-module-5-databases	13 Feb 2022 2...	idanylyuk	f0848977
DEV-50-module-5-databases origin	hw5	13 Feb 2022 2...	idanylyuk	8a5d63b8
	Merge pull request #9 from idanylyuk/DEV-49-module-4-packaging	13 Feb 2022 2...	idanylyuk	ca004d23
DEV-49-module-4-packaging origin	hw4	13 Feb 2022 2...	idanylyuk	39dd1044
	Merge pull request #8 from idanylyuk/DEV-48-module-3-networking	13 Feb 2022 2...	idanylyuk	d504d118
DEV-48-module-3-networking origin	hw3	13 Feb 2022 2...	idanylyuk	66cf7a51
	Merge pull request #7 from idanylyuk/DEV-47-module-2-text	13 Feb 2022 2...	idanylyuk	7ae9eafa
DEV-47-module-2-text origin	hw2	13 Feb 2022 2...	idanylyuk	9635289c
	Merge pull request #6 from idanylyuk/DEV-46-module-1-operating-system-interfaces	13 Feb 2022 2...	idanylyuk	ca2c19e5
DEV-46-module-1-operating-system-interfaces origin	hw1	13 Feb 2022 2...	idanylyuk	7f68fd13
	HW1	13 Feb 2022 2...	idanylyuk	34b4c3e2
	Git clear	13 Feb 2022 1...	idanylyuk	bca86869

Ivan Danyliuk

Markdown



```
# Geocitizen project

### 1. Requirements to Host PC
`Host PC with Oracle VirtualBox and SSH client installed.`

### 2. Create 2 VitrualBox VMs:
**VM1 (Apache Tomcat Server)** (1-2 Gb RAM):

***Ubuntu 16.04 Server*** [Configuration details]
(TomcatServer.md)

- Open SSH Server
- Oracle Java SDK 8
- Apache Tomcat/9.0.58
- Apache Maven 3.3.9

**VM2 (PostgreSQL Database Server)** (1 Gb RAM):

***CentOS 7.9.2009*** [Configuration details](PostgreSQL.md)

- Open SSH Server
- PostgreSQL 9.2.24
### 3. Application fixing / deployment

#### - Released on GitHub application have some [bugs]
(AppBugs.md). They are fixed automatically by deploy
application script (see below).

#### - Get IP addresses of Apache Tomcat Server and PostgreSQL
Database Server. You may use commands

`$ ifconfig -a`
```

Geocitizen project

1. Requirements to Host PC

Host PC with Oracle VirtualBox and SSH client installed.

2. Create 2 VitrualBox VMs:

VM1 (Apache Tomcat Server) (1-2 Gb RAM):

Ubuntu 16.04 Server [Configuration details](#)

- Open SSH Server
- Oracle Java SDK 8
- Apache Tomcat/9.0.58
- Apache Maven 3.3.9

VM2 (PostgreSQL Database Server) (1 Gb RAM):

CentOS 7.9.2009 [Configuration details](#)

- Open SSH Server
- PostgreSQL 9.2.24

3. Application fixing / deployment

- Released on GitHub application have some [bugs](#). They are fixed automatically by deploy application script (see below).
- Get IP addresses of Apache Tomcat Server and PostgreSQL Database Server. You may use commands

```
$ ifconfig -a
```

Manual deploy Geocitizen Application

Deploy Application
on 2 Virtualbox VMs

GitHub repository:

<https://github.com/mentorchita/Geocit134>

ch-058, geocitizen

build and deploy (ubuntu16, git2, maven3, tomcat9)

1. `git clone https://github.com/nromanen/Ch-058.git; cd Ch-058`
2. in config file `~/Ch-058/src/main/resources/application.properties` you might want to edit following properties
 - `front.url` - front url
 - `db.url` - db uri (**db must be created manually**)
 - `db.username` & `db.password` - db credentials
3. `mvn install && mv target/citizen.war /usr/share/tomcat9/webapps/ && /usr/share/tomcat9/bin/startup.sh`
4. e.g. <http://localhost:8080/citizen/>

Ivan Danyliuk

Oracle VM VirtualBox



Host operating system (host OS):

Windows, Mac OS X, Linux, and Oracle Solaris hosts.

Guest operating system (guest OS):

x86/x64 OS such as DOS, Windows, OS/2, FreeBSD, and OpenBSD.

Guest Additions:

special software packages installed inside a VM to improve performance of the guest OS and to add extra features.

Geocitizen



Clone correct repository

```
git clone
https://github.com/
mentorchita/Geocit1
34.git
```

Oracle VM VirtualBox

Create VM (4G RAM)

OS Ubuntu 16 Desktop

Install and configure

OpenSSH Server

Oracle Java SDK 8

Apache Tomcat 9.0.58

Apache Maven 3.3.9

PostgreSQL 9.2.24

Deploy App to localhost on Ubuntu16 VM

Fix bugs

old java repository bugs

developers bugs

Add correct

server's ip-address

database and gmail credentials

Create and configure 2 connected VMs

VM1 (Apache Tomcat Server) (1-2 Gb RAM):

Ubuntu 16.04 Server

OpenSSH Server,
Oracle Java SDK 8,
Apache Tomcat/9.0.58,
Apache Maven 3.3.9

VM2 (PostgreSQL Database Server) (1 Gb RAM):

CentOS 7.9.2009

OpenSSH Server,
PostgreSQL 9.2.24

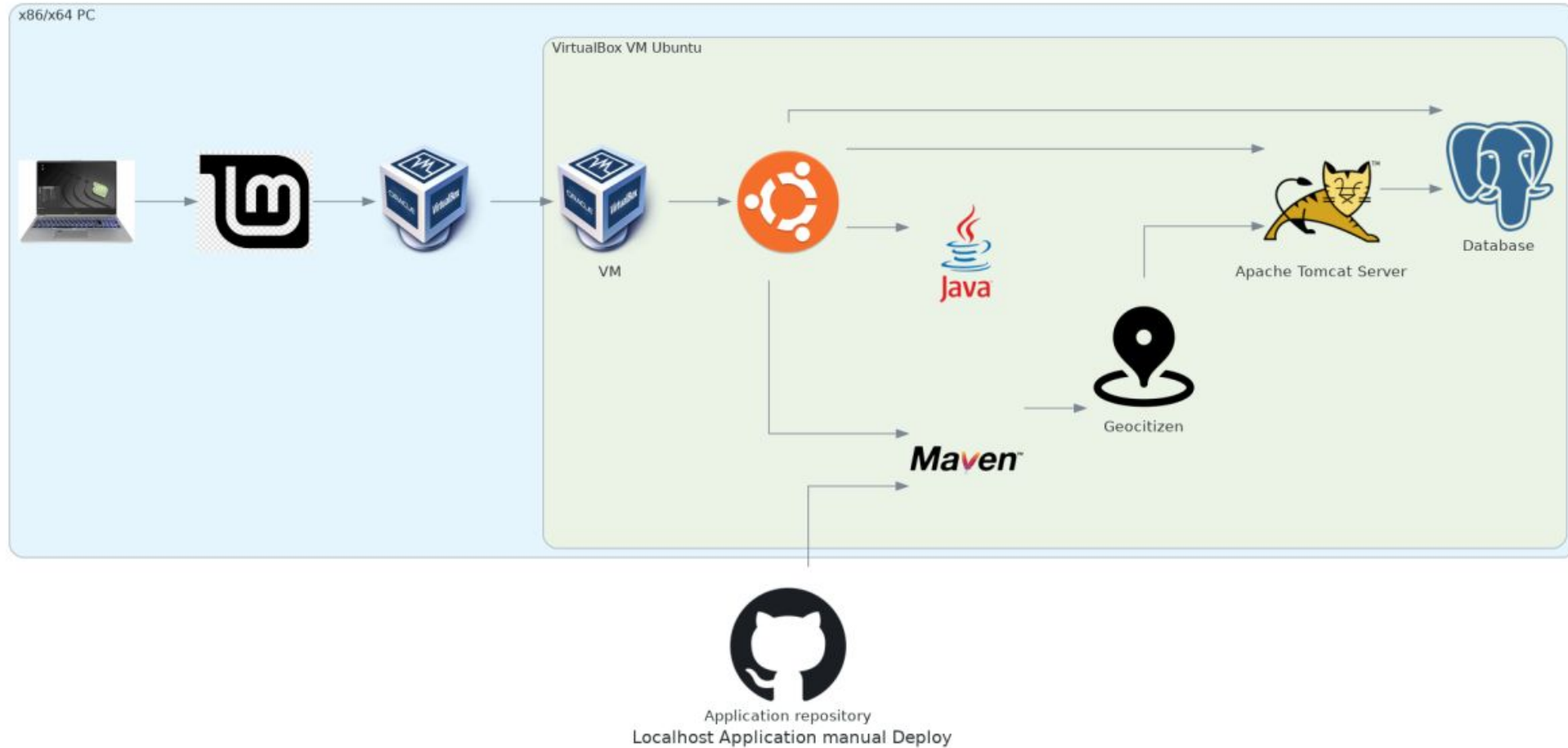
Create readme.md

Create readme

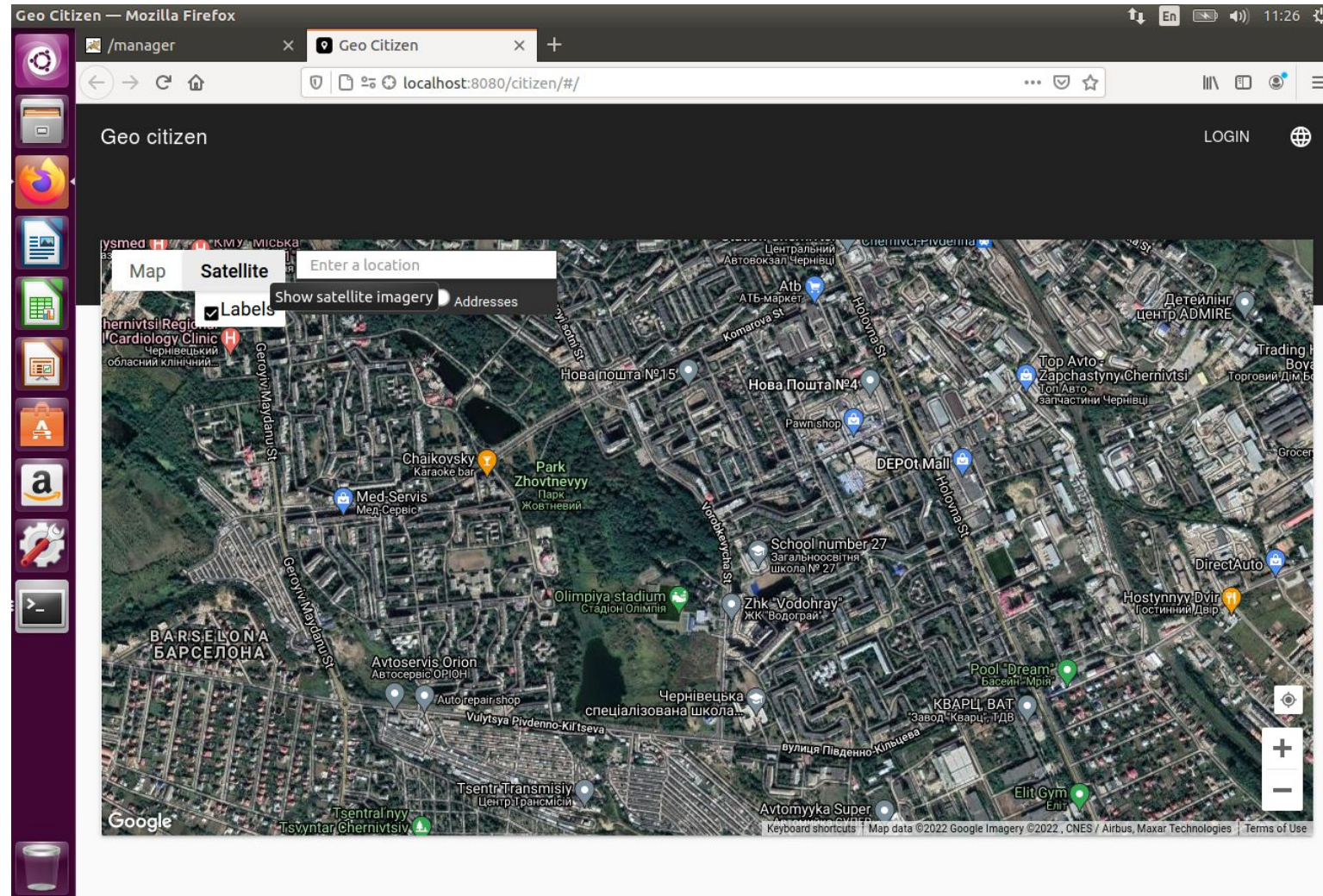
describe the process of
manual deploy
application on GitHub
Markdown

Ivan Danyliuk

Deploy App to localhost on Ubuntu 16 VM



Run App on localhost inside Ubuntu 16 VM



Ivan Danyliuk

Bugs and fixes to Geocitizen

1. Repositories changed from date of release

replace `http://repo.spring.io` with `https://repo.spring.io`

2. Remove unused local repository section

```
<distributionManagement> ... </distributionManagement>
```

3. Remove `localhost` adresse from project

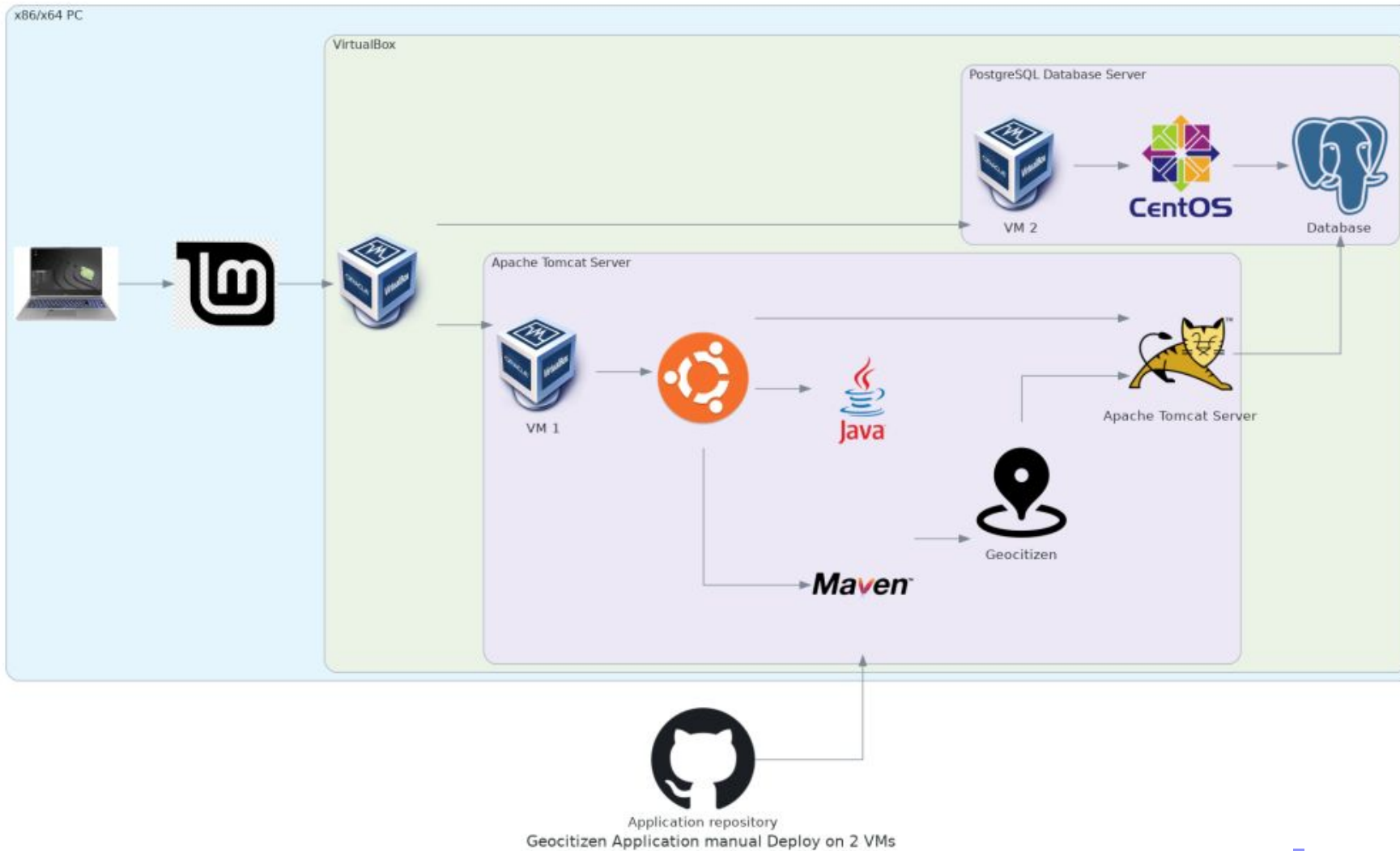
4. Add real Apache Tomcat Application Server and PostgreSQL Database Server ip-adresse.

5. Add database credentials.

6. Add Gmail credential fixes.

7. Correct favicon path with correct one.

Deploy App on 2 VMs



Ivan Danyliuk

Deploy application script

```
#remove old files
rm -rf Geocit134

# Clone git repository
git clone https://github.com/mentorchita/Geocit134.git
cd Geocit134

server_ip='10.1.1.112'
db_server_ip='10.1.1.110'
db_name='db_name'
db_user='db_user'
db_password='db_password'
gmail_user='gmail_user'
gmail_password='gmail_user_password'
#fix pom.xml file bugs
find . -type f -name "pom.xml" -exec sed -i 's/http:\\\\repo.spring.io/https:\\\\repo.spring.io/g' {} +
find . -type f -name "pom.xml" -exec sed -i 's/<distributionManagement>/<!--<distributionManagement>/g' {} +
find . -type f -name "pom.xml" -exec sed -i 's/<\\/distributionManagement>/<\\/distributionManagement>-->/g' {} +

#fix database application connection
find . -type f -name "application.properties" -exec sed -i "s/http:\\\\localhost/http:\\\\$server_ip/g" {} +
find . -type f -name "application.properties" -exec sed -i "s/postgresql:\\\\localhost:5432\\ss_demo_1/postgresql:\\\\$db_server_ip:5432\\$db_name/g" {} +
find . -type f -name "application.properties" -exec sed -i "s/postgresql:\\\\35.204.28.238:5432\\ss_demo_1/postgresql:\\\\$db_server_ip:5432\\$db_name/g" {} +
find . -type f -name "application.properties" -exec sed -i "s/username=postgres/username=$db_user/g" {} +
find . -type f -name "application.properties" -exec sed -i "s/password=postgres/password=$db_password/g" {} +

# fix js localhost bugs
find . -type f -name "*.js" -exec sed -i "s/localhost:8080/$server_ip:8080/g" {} +

# fix e-mail data
find . -type f -name "application.properties" -exec sed -i "s/ssgeocitizen/$gmail_user/g" {} +
find . -type f -name "application.properties" -exec sed -i "s/password=softserve/password=$gmail_password/g" {} +

#fix favicon
find . -type f -name "index.html" -exec sed -i "s\\/src\\/assets\\/\\.\\.\\/static/g" {} +

#build project
mvn install

#deploy project
sudo mv target/citizen.war /opt/tomcat/webapps
```

Ivan Danyliuk

Deploy App on 2 VMs with script

```
test@TomcatServer:~$ ./deploy
Cloning into 'Geocit134'...
remote: Enumerating objects: 408, done.
remote: Counting objects: 100% (408/408), done.
remote: Compressing objects: 100% (309/309), done.
remote: Total 408 (delta 77), reused 403 (delta 76), pack-reused 0
Receiving objects: 100% (408/408), 1.72 MiB | 852.00 KiB/s, done.
Resolving deltas: 100% (77/77), done.
Checking connectivity... done.
```

```
[INFO] Installing /home/test/Geocit134/pom.xml to /home/test/.m2/repository/com/
softserveinc/geo-citizen/1.0.5-SNAPSHOT/geo-citizen-1.0.5-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 14.812 s
[INFO] Finished at: 2022-02-14T12:20:25+02:00
[INFO] Final Memory: 41M/219M
[INFO] -----
[sudo] password for test:
test@TomcatServer:~$
```

```
1 [|||||86.8%] Tasks: 27, 43 thr: 4 running
2 [|||||100.0%] Load average: 0.75 0.38 0.17
Mem[|||||770M/1.95G] Uptime: 00:05:17
Swp[|||||0K/975M]

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
 1024 tomcat      20    0 3502M  710M 22392 S 200. 35.5  1:25.84 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1132 tomcat      20    0 3502M  710M 22392 R 166. 35.5  0:37.48 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1579 tomcat      20    0 3502M  710M 22392 R 122. 35.5  0:15.11 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1133 tomcat      20    0 3502M  710M 22392 S 34.3 35.5  0:07.33 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1120 tomcat      20    0 3502M  710M 22392 S  2.3 35.5  0:01.01 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1119 tomcat      20    0 3502M  710M 22392 S  1.1 35.5  0:01.11 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1127 tomcat      20    0 3502M  710M 22392 S  1.1 35.5  0:00.81 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1788 test        20    0 26056  3840  3128 R  1.1  0.2  0:00.11 htop
 1026 root         10  -10   5716  3512  2432 S  0.0  0.2  0:00.10 /sbin/iscsid
 1789 tomcat      20    0 3502M  710M 22392 S  0.0 35.5  0:00.01 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1135 tomcat      20    0 3502M  710M 22392 S  0.0 35.5  0:00.50 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1575 tomcat      20    0 3502M  710M 22392 S  0.0 35.5  0:00.08 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1590 tomcat      20    0 3502M  710M 22392 S  0.0 35.5  0:00.05 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
 1128 tomcat      20    0 3502M  710M 22392 S  0.0 35.5  0:00.03 /usr/lib/jvm/jdk1.8.0_321/jre/bin/jav
```

Ivan Danyliuk

Application on host OS via Firefox browser

