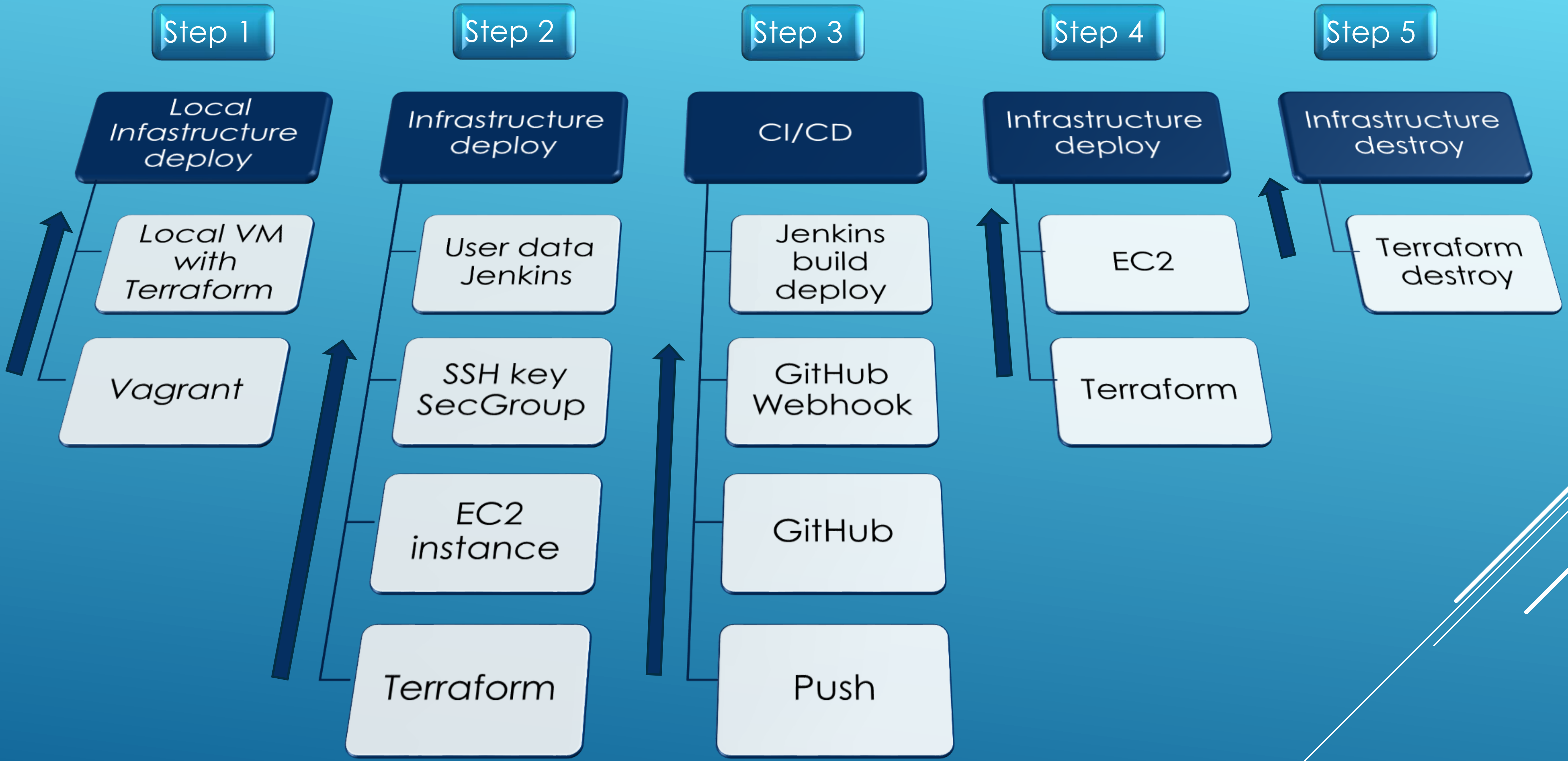
OLEXANDR MARCHENKO

DATE OF BIRTHDAY 28/11/75
HIGH EDUCATION
CIVIL ENGINEER
EXP. IN IT: BEGINNER (CMS - WORDPRESS, JOOMLA, ZEBRA)

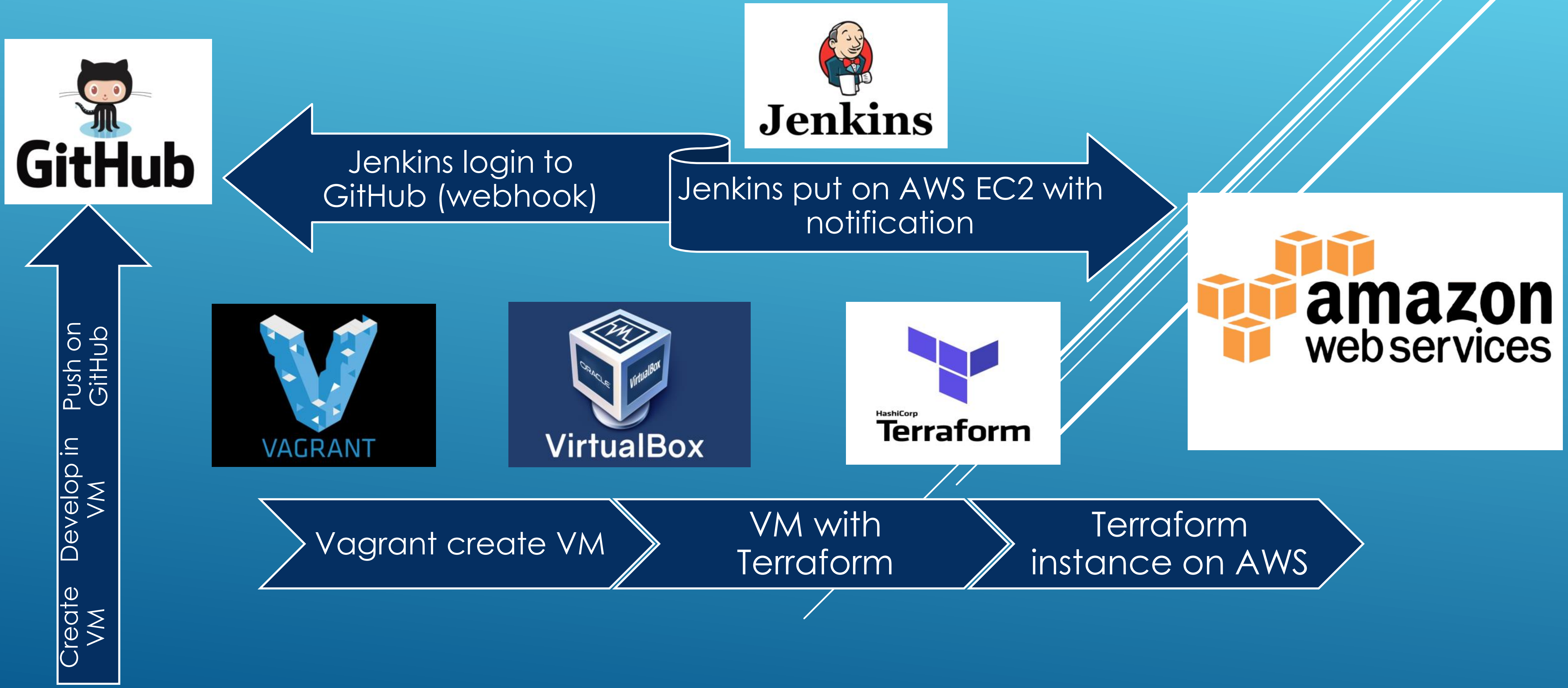
I WANT TO REACH NEW HEIGHTS AND START WORKING IN EPAM

ABOUT ME


Step 1
Local Infastructure deploy
Local VM with Terraform
Vagrant
Step 2
Infrastructure deploy
User data Jenkins
SSH key SecGroup
EC2 instance
Terraform
Step 3
CI/CD
Jenkins build deploy
GitHub Webhook
GitHub
Push
Step 4
Infrastructure deploy
EC2
Terraform
Step 5
Infrastructure destroy
Terraform destroy

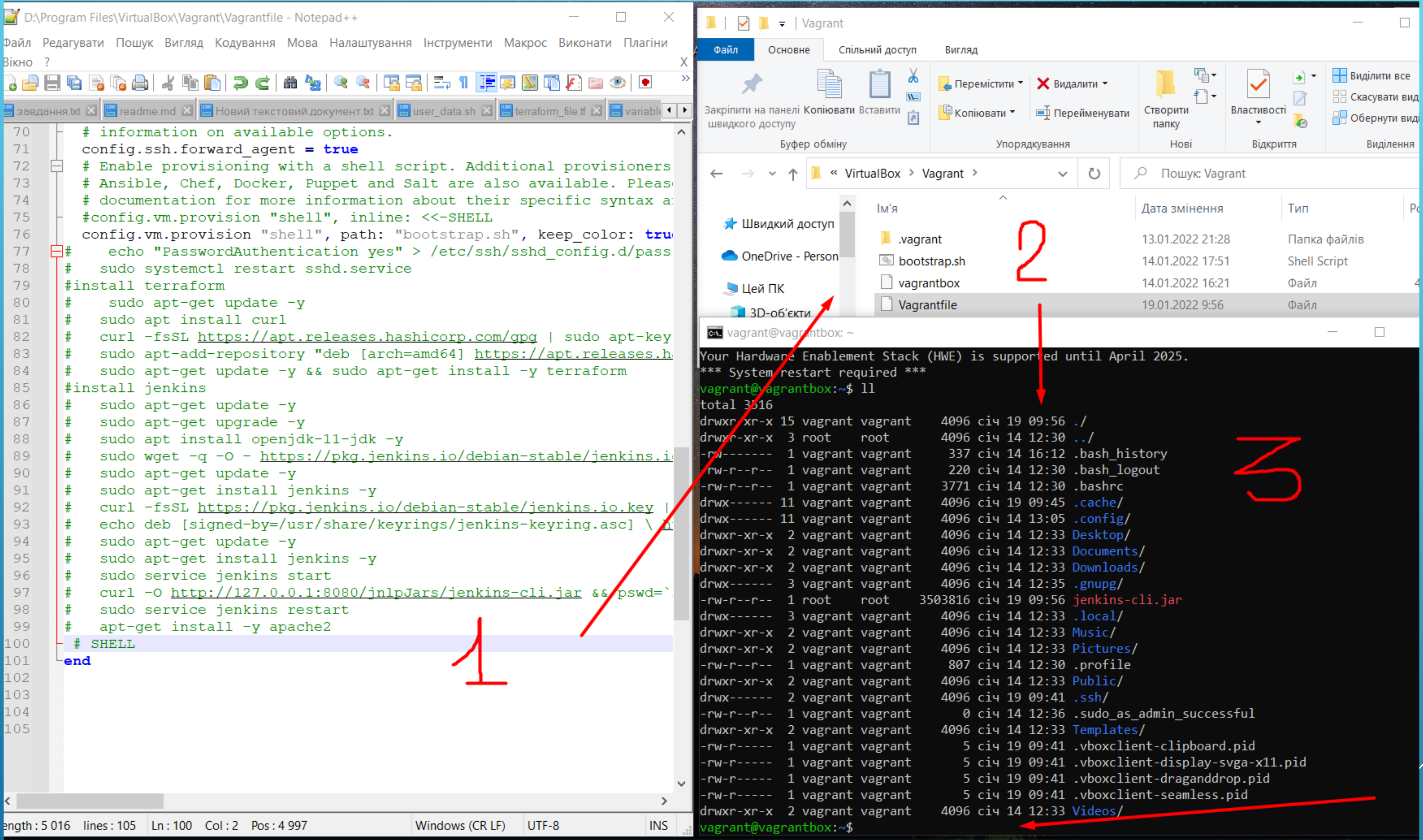
DEPLOY TELEGRAM BOT

For my project I used: Github, Vagrant, Virtualbox, Terraform, Jenkins, AWS



- I used Vagarnt for creation VM Ubuntu (my own image) with Terraform (step1) on local host. Then I create instance Ubuntu on AWS with Jenkins (step2).
- After that I configurate Jenkins (step3). Used next plugin: CloudBees AWS Credentials, SSH agent, Git plugin, Github plugin, Pipeline etc.
- Developer with Windows OS and installed Bush plus credential for GitHub. He edits python scrypt and push on GitHub. Then Github sends webhook for Jenkins (step 3). Jenkins used Jenkinsfile from Github and performs jobs from this file. (take, test, create artefact, create artefact version, deploy , notification.) I created daemon, and a daemon was used to run the application.

REALIZATION



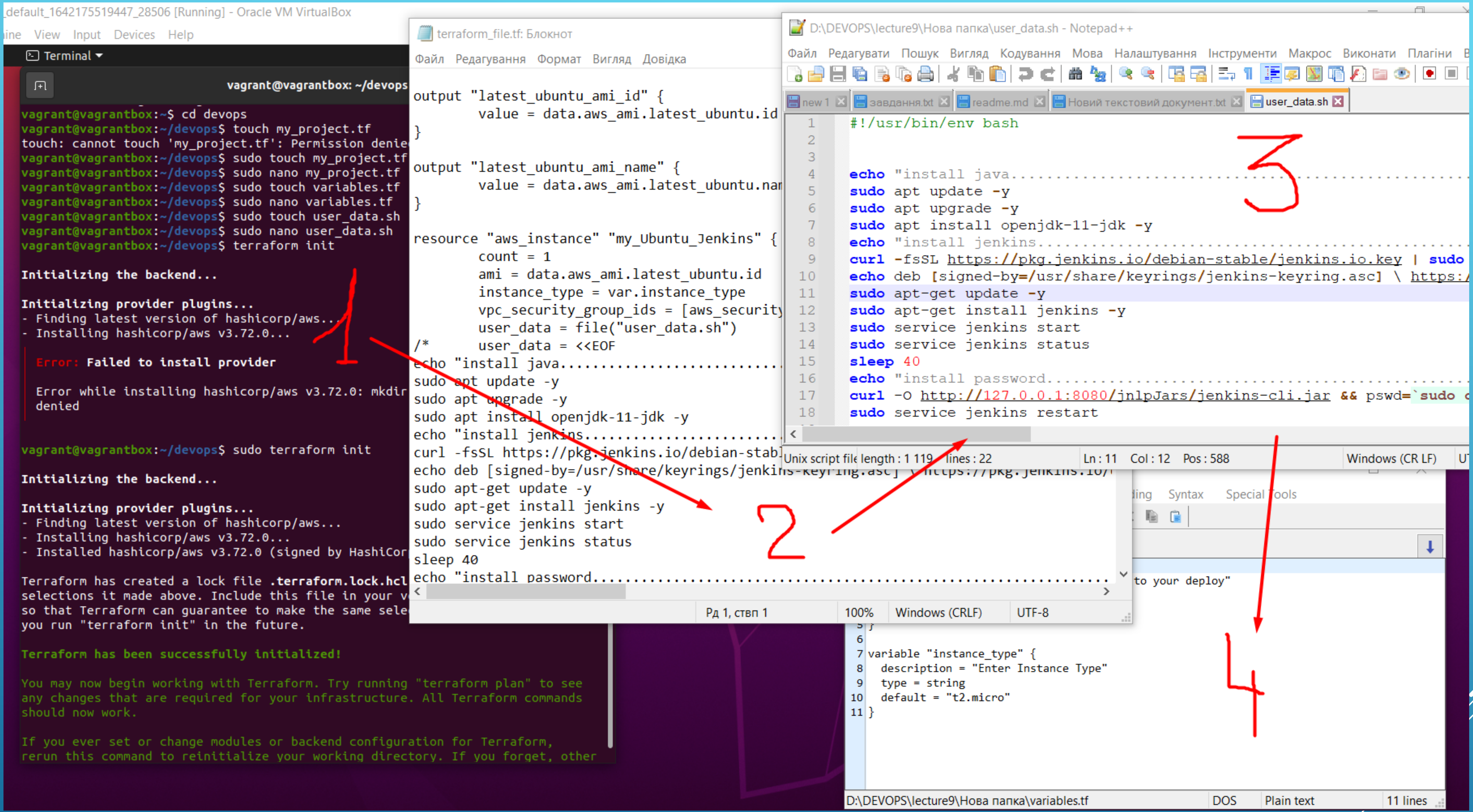
```
  # information on available options.
  config.ssh.forward_agent = true
  # Enable provisioning with a shell script. Additional provisioners
  # Ansible, Chef, Docker, Puppet and Salt are also available. Pleas
  # documentation for more information about their specific syntax a
  #config.vm.provision "shell", inline: <<-SHELL
  config.vm.provision "shell", path: "bootstrap.sh", keep_color: tru
#    echo "PasswordAuthentication yes" > /etc/ssh/sshd_config.d/pass
#   sudo systemctl restart sshd.service
#install terraform
#    sudo apt-get update -y
#   sudo apt install curl
#   curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key
#   sudo apt-add-repository "deb [arch=amd64] https://apt.releases.h
#   sudo apt-get update -y && sudo apt-get install -y terraform
#install jenkins
#   sudo apt-get update -y
#   sudo apt-get upgrade -y
#   sudo apt install openjdk-11-jdk -y
#   sudo wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.i
#   sudo apt-get update -y
#   sudo apt-get install jenkins -y
#   curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key |
#   echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \ h
#   sudo apt-get update -y
#   sudo apt-get install jenkins -y
#   sudo service jenkins start
#   curl -O http://127.0.0.1:8080/jnlpJars/jenkins-cli.jar && pswd=`
#   sudo service jenkins restart
#   apt-get install -y apache2
 # SHELL
end
```

ength : 5 016   lines : 105   Ln : 100   Col : 2   Pos : 4 997   Windows (CR LF)   UTF-8   INS

Ім'я	Дата змінення	Тип
.vagrant	13.01.2022 21:28	Папка файлів
bootstrap.sh	14.01.2022 17:51	Shell Script
vagrantbox	14.01.2022 16:21	Файл
Vagrantfile	19.01.2022 9:56	Файл

```
Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
vagrant@vagrantbox:~$ ll
total 3516
drwxr-xr-x 15 vagrant vagrant    4096 січ 19 09:56 ./
drwxr-xr-x  3 root    root       4096 січ 14 12:30 ../
-rw-------  1 vagrant vagrant     337 січ 14 16:12 .bash_history
-rw-r--r--  1 vagrant vagrant     220 січ 14 12:30 .bash_logout
-rw-r--r--  1 vagrant vagrant    3771 січ 14 12:30 .bashrc
drwx------ 11 vagrant vagrant    4096 січ 19 09:45 .cache/
drwx------ 11 vagrant vagrant    4096 січ 14 13:05 .config/
drwxr-xr-x  2 vagrant vagrant    4096 січ 14 12:33 Desktop/
drwxr-xr-x  2 vagrant vagrant    4096 січ 14 12:33 Documents/
drwxr-xr-x  2 vagrant vagrant    4096 січ 14 12:33 Downloads/
drwx------  3 vagrant vagrant    4096 січ 14 12:35 .gnupg/
-rw-r--r--  1 root    root    3503816 січ 19 09:56 jenkins-cli.jar
drwx------  3 vagrant vagrant    4096 січ 14 12:33 .local/
drwxr-xr-x  2 vagrant vagrant    4096 січ 14 12:33 Music/
drwxr-xr-x  2 vagrant vagrant    4096 січ 14 12:33 Pictures/
-rw-r--r--  1 vagrant vagrant     807 січ 14 12:30 .profile
drwxr-xr-x  2 vagrant vagrant    4096 січ 14 12:33 Public/
drwx------  2 vagrant vagrant    4096 січ 19 09:41 .ssh/
-rw-r--r--  1 vagrant vagrant       0 січ 14 12:36 .sudo_as_admin_successful
drwxr-xr-x  2 vagrant vagrant    4096 січ 14 12:33 Templates/
-rw-r-----  1 vagrant vagrant       5 січ 19 09:41 .vboxclient-clipboard.pid
-rw-r-----  1 vagrant vagrant       5 січ 19 09:41 .vboxclient-display-svga-x11.pid
-rw-r-----  1 vagrant vagrant       5 січ 19 09:41 .vboxclient-draganddrop.pid
-rw-r-----  1 vagrant vagrant       5 січ 19 09:41 .vboxclient-seamless.pid
drwxr-xr-x  2 vagrant vagrant    4096 січ 14 12:33 Videos/
vagrant@vagrantbox:~$
```

CREATED VM WITH VAGRANT THEN VAGRANT UP. AFTER THEN INSTALLED TERRAFORM.


default_1642175519447_28506 [Running] - Oracle VM VirtualBox
ine View Input Devices Help
Terminal
vagrant@vagrantbox: ~/devops
terraform_file.tf: Блокнот
Файл Редагування Формат Вигляд Довідка
D:\DEVOPS\lecture9\Нова папка\user_data.sh - Notepad++
Файл Редагувати Пошук Вигляд Кодування Мова Налаштування Інструменти Макрос Виконати Плагіни
new 1
завдання.txt
readme.md
Новий текстовий документ.txt
user_data.sh
1
2
3
4

```
vagrant@vagrantbox:~$ cd devops
vagrant@vagrantbox:~/devops$ touch my_project.tf
touch: cannot touch 'my_project.tf': Permission denie
vagrant@vagrantbox:~/devops$ sudo touch my_project.tf
vagrant@vagrantbox:~/devops$ sudo nano my_project.tf
vagrant@vagrantbox:~/devops$ sudo touch variables.tf
vagrant@vagrantbox:~/devops$ sudo nano variables.tf
vagrant@vagrantbox:~/devops$ sudo touch user_data.sh
vagrant@vagrantbox:~/devops$ sudo nano user_data.sh
vagrant@vagrantbox:~/devops$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.72.0...

  Error: Failed to install provider

  Error while installing hashicorp/aws v3.72.0: mkdir
  denied

vagrant@vagrantbox:~/devops$ sudo terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.72.0...
- Installed hashicorp/aws v3.72.0 (signed by HashiCor

Terraform has created a lock file .terraform.lock.hcl
selections it made above. Include this file in your v
so that Terraform can guarantee to make the same sele
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
```

```
output "latest_ubuntu_ami_id" {
        value = data.aws_ami.latest_ubuntu.id
}

output "latest_ubuntu_ami_name" {
        value = data.aws_ami.latest_ubuntu.nar
}

resource "aws_instance" "my_Ubuntu_Jenkins" {
        count = 1
        ami = data.aws_ami.latest_ubuntu.id
        instance_type = var.instance_type
        vpc_security_group_ids = [aws_security
        user_data = file("user_data.sh")
/*      user_data = <<EOF
echo "install java.............................
sudo apt update -y
sudo apt upgrade -y
sudo apt install openjdk-11-jdk -y
echo "install jenkins..........................
curl -fsSL https://pkg.jenkins.io/debian-stabl
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \ https://pkg.jenkins.io/
sudo apt-get update -y
sudo apt-get install jenkins -y
sudo service jenkins start
sudo service jenkins status
sleep 40
echo "install password.................................................................
```

Рд 1, стпв 1 | 100% | Windows (CRLF) | UTF-8

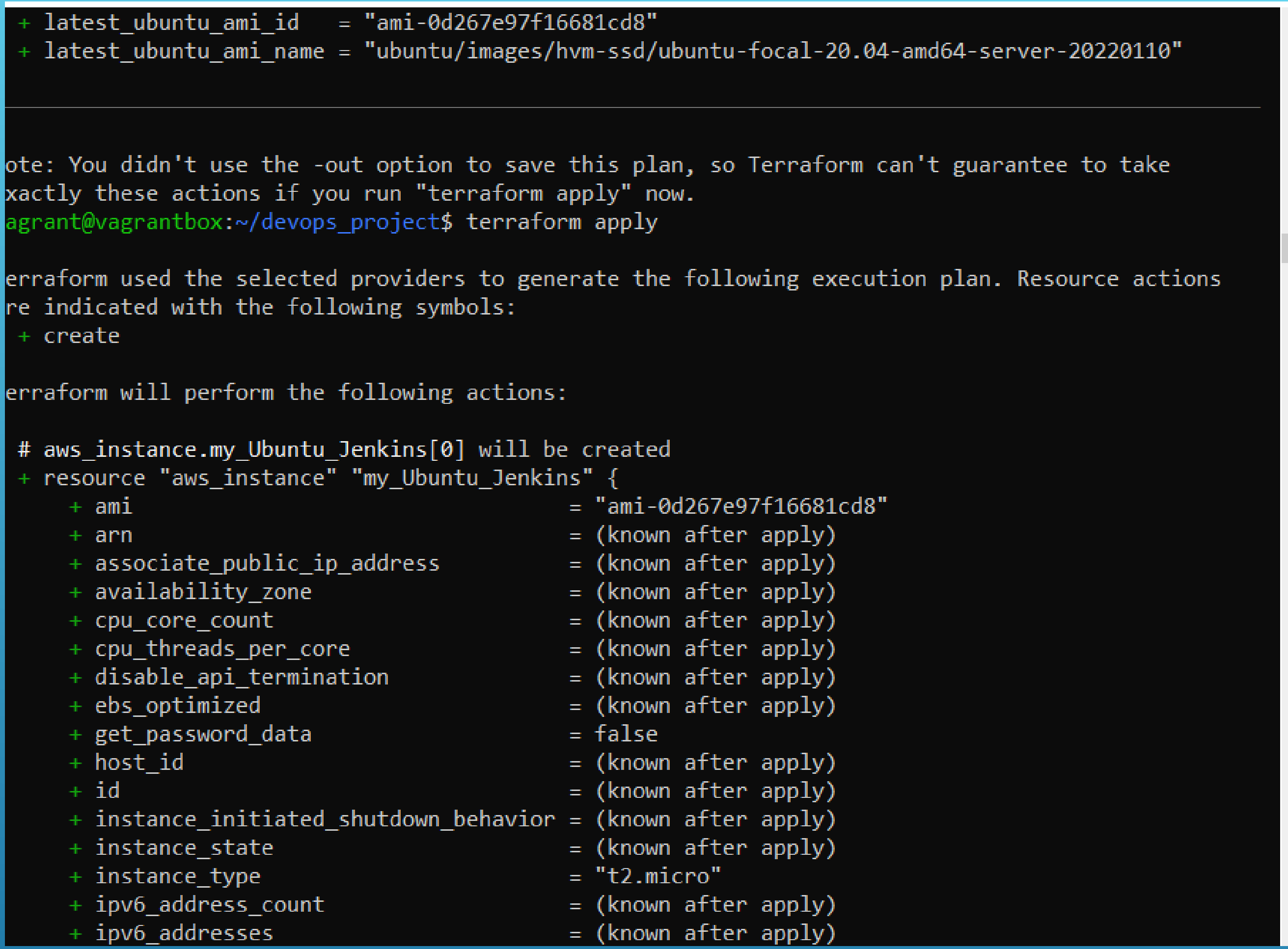
```
#!/usr/bin/env bash


echo "install java...............................................
sudo apt update -y
sudo apt upgrade -y
sudo apt install openjdk-11-jdk -y
echo "install jenkins............................................
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \ https:/
sudo apt-get update -y
sudo apt-get install jenkins -y
sudo service jenkins start
sudo service jenkins status
sleep 40
echo "install password...........................................
curl -O http://127.0.0.1:8080/jnlpJars/jenkins-cli.jar && pswd=`sudo c
sudo service jenkins restart
```

Unix script file length : 1 119 lines : 22 | Ln : 11 Col : 12 Pos : 588 | Windows (CR LF)

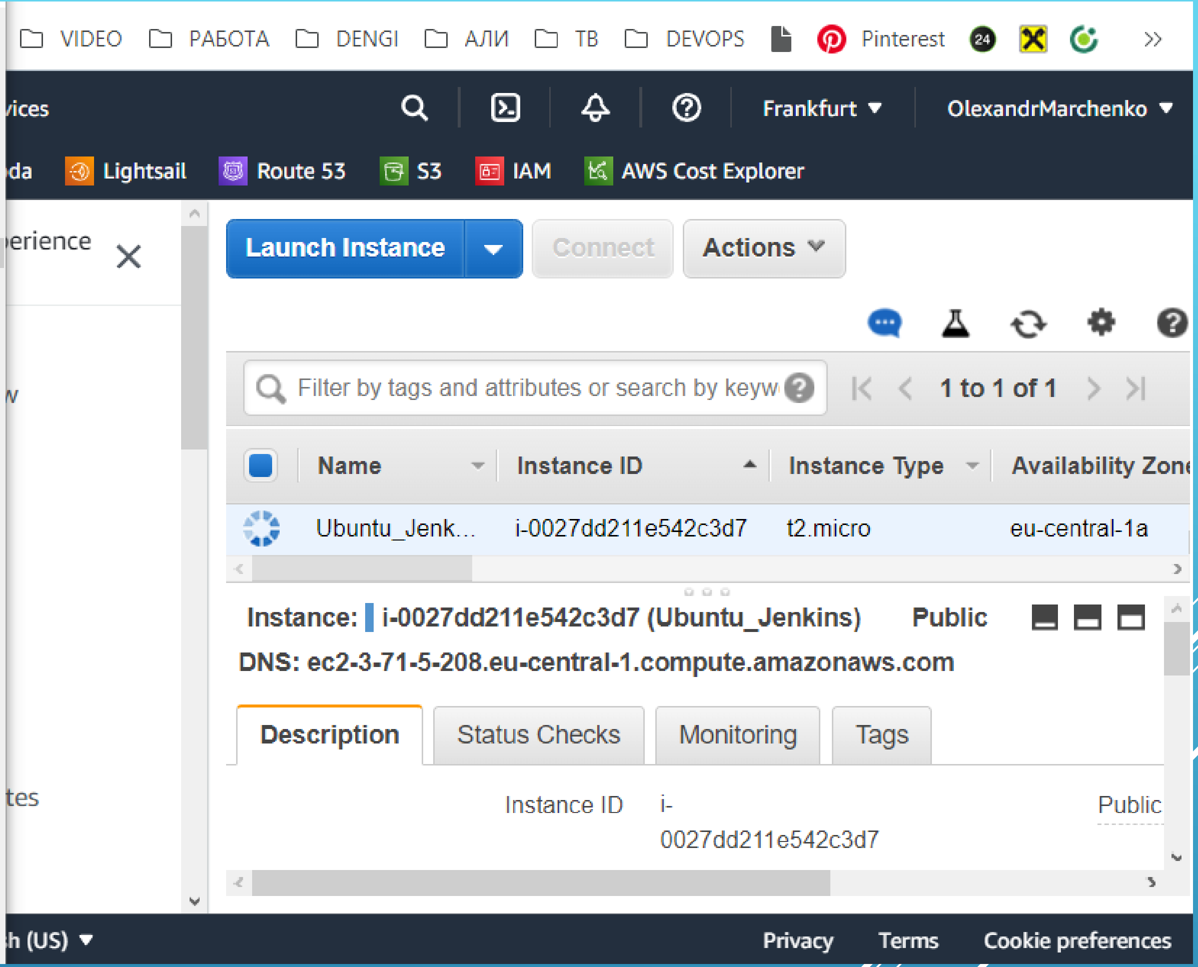
```
to your deploy"
}

variable "instance_type" {
  description = "Enter Instance Type"
  type = string
  default = "t2.micro"
}
```

D:\DEVOPS\lecture9\Нова папка\variables.tf | DOS | Plain text | 11 lines

CREATED (VAGRANT SSH) PROJECT FOLDER AND TERRAFORM FILES



```
 + latest_ubuntu_ami_id   = "ami-0d267e97f16681cd8"
 + latest_ubuntu_ami_name = "ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-20220110"

ote: You didn't use the -out option to save this plan, so Terraform can't guarantee to take
xactly these actions if you run "terraform apply" now.
agrant@vagrantbox:~/devops_project$ terraform apply

erraform used the selected providers to generate the following execution plan. Resource actions
re indicated with the following symbols:
 + create

erraform will perform the following actions:

 # aws_instance.my_Ubuntu_Jenkins[0] will be created
 + resource "aws_instance" "my_Ubuntu_Jenkins" {
     + ami                                  = "ami-0d267e97f16681cd8"
     + arn                                  = (known after apply)
     + associate_public_ip_address          = (known after apply)
     + availability_zone                    = (known after apply)
     + cpu_core_count                       = (known after apply)
     + cpu_threads_per_core                 = (known after apply)
     + disable_api_termination              = (known after apply)
     + ebs_optimized                        = (known after apply)
     + get_password_data                    = false
     + host_id                              = (known after apply)
     + id                                   = (known after apply)
     + instance_initiated_shutdown_behavior = (known after apply)
     + instance_state                       = (known after apply)
     + instance_type                        = "t2.micro"
     + ipv6_address_count                   = (known after apply)
     + ipv6_addresses                       = (known after apply)
```



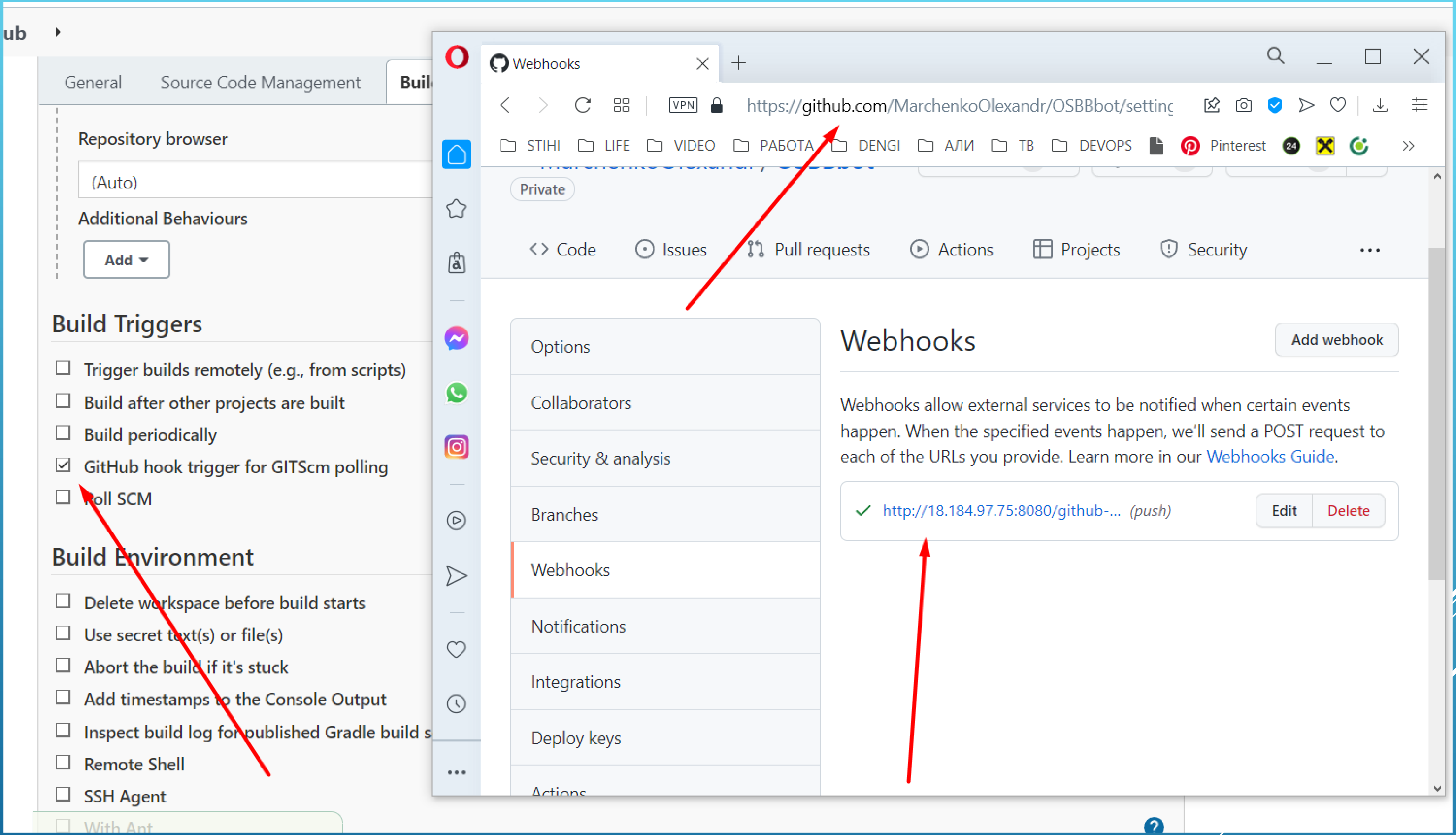
Name	Instance ID	Instance Type	Availability Zone
Ubuntu_Jenk...	i-0027dd211e542c3d7	t2.micro	eu-central-1a

Instance: i-0027dd211e542c3d7 (Ubuntu_Jenkins) Public DNS: ec2-3-71-5-208.eu-central-1.compute.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID i-0027dd211e542c3d7 Public

USED TERRAFORM FOR CREATE INSTANCE ON AWS. INSTANCE INCLUDE JENKINS


General | Source Code Management

Repository browser

(Auto)

Additional Behaviours

Add

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☐ Build periodically
- ☑ GitHub hook trigger for GITScm polling
- ☐ Poll SCM

Build Environment

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Abort the build if it's stuck
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published Gradle build s
- ☐ Remote Shell
- ☐ SSH Agent
- ☐ With Ant

https://github.com/MarchenkoOlexandr/OSBBbot/setting

Private

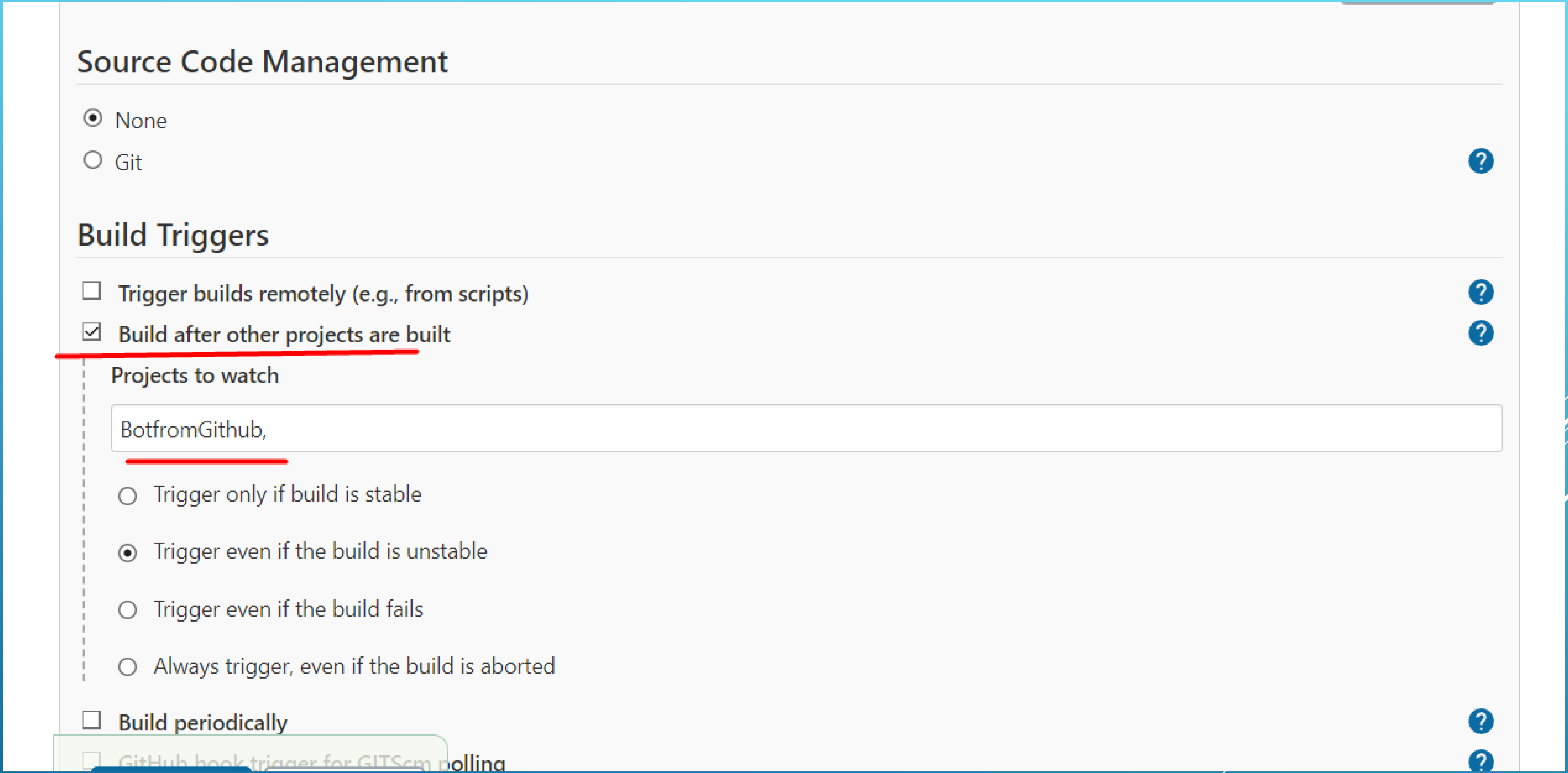
Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://18.184.97.75:8080/github-... (push) Edit Delete

JENKINS PULL FROM GITHUB WITH WEBHOOK, SAVE ARTEFACT AND DEPLOY TO SERVER



Source Code Management

- (●) None
- ( ) Git

Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts)
- [✓] Build after other projects are built

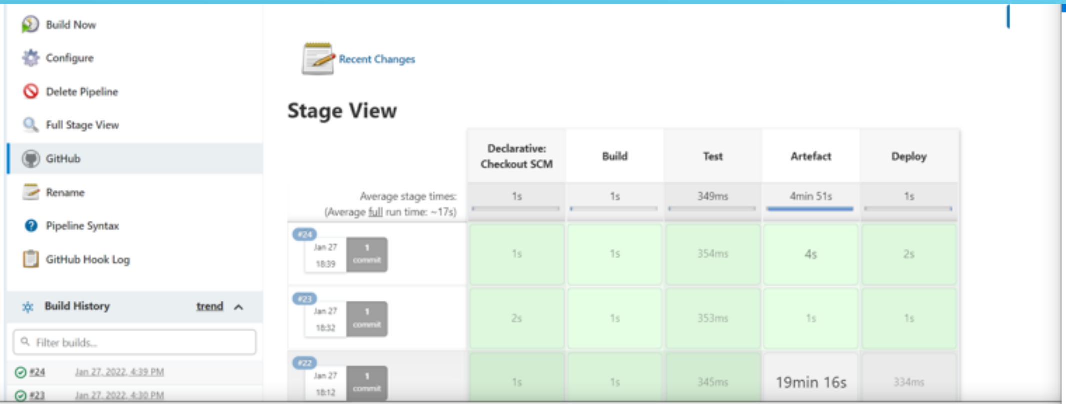
Projects to watch

BotfromGithub,

- ( ) Trigger only if build is stable
- (●) Trigger even if the build is unstable
- ( ) Trigger even if the build fails
- ( ) Always trigger, even if the build is aborted

- [ ] Build periodically
- [ ] GitHub hook trigger for GITScm polling

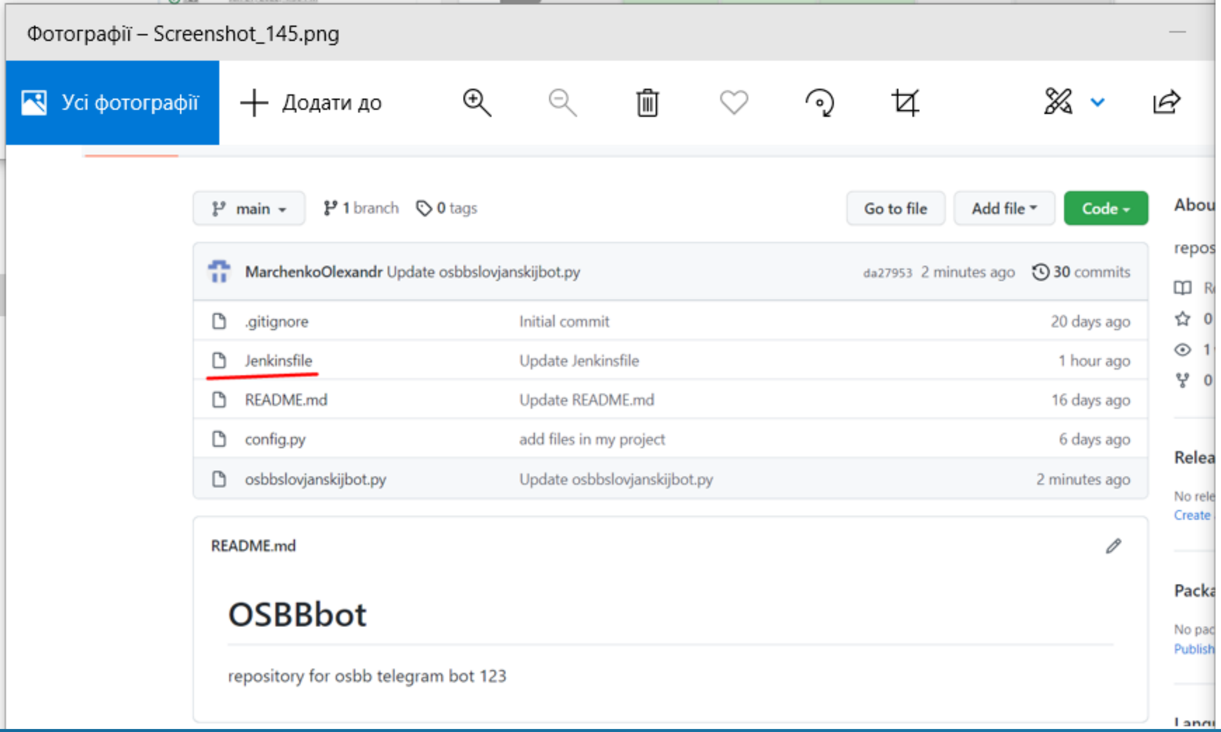
SIMPLE PIPELINE



Stage View

	Declarative: Checkout SCM	Build	Test	Artefact	Deploy
Average stage times: (Average full run time: ~17s)	1s	1s	349ms	4min 51s	1s
#24 Jan 27 18:39 1 commit	1s	1s	354ms	4s	2s
#23 Jan 27 18:32 1 commit	2s	1s	353ms	1s	1s
#22 Jan 27 18:12 1 commit	1s	1s	345ms	19min 16s	334ms

#24 Jan 27, 2022, 4:39 PM
#23 Jan 27, 2022, 4:30 PM



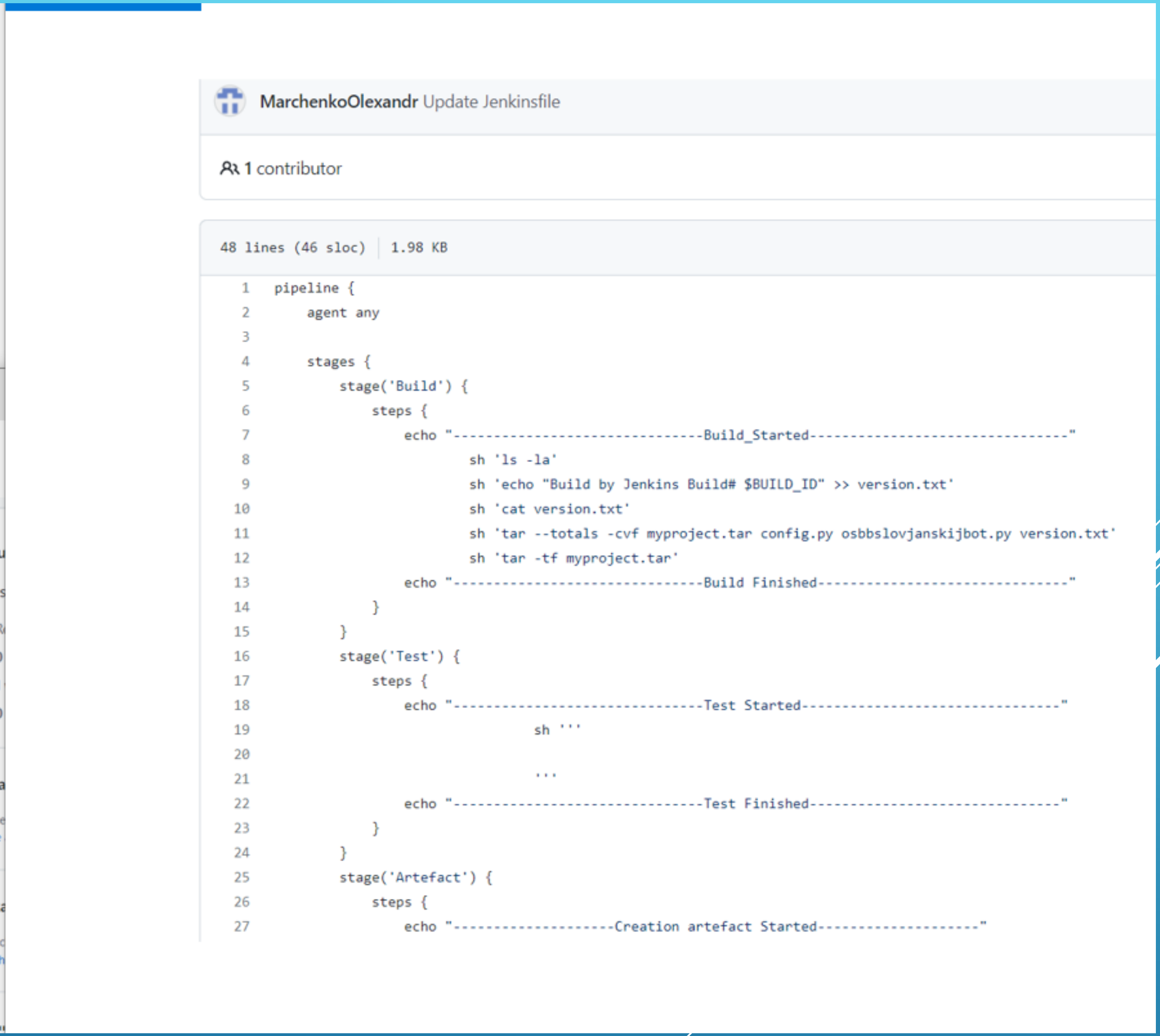
MarchenkoOlexandr Update osbbslovjanskijbot.py da27953 2 minutes ago 30 commits

.gitignore	Initial commit	20 days ago
Jenkinsfile	Update Jenkinsfile	1 hour ago
README.md	Update README.md	16 days ago
config.py	add files in my project	6 days ago
osbbslovjanskijbot.py	Update osbbslovjanskijbot.py	2 minutes ago

README.md

OSBBbot

repository for osbb telegram bot 123

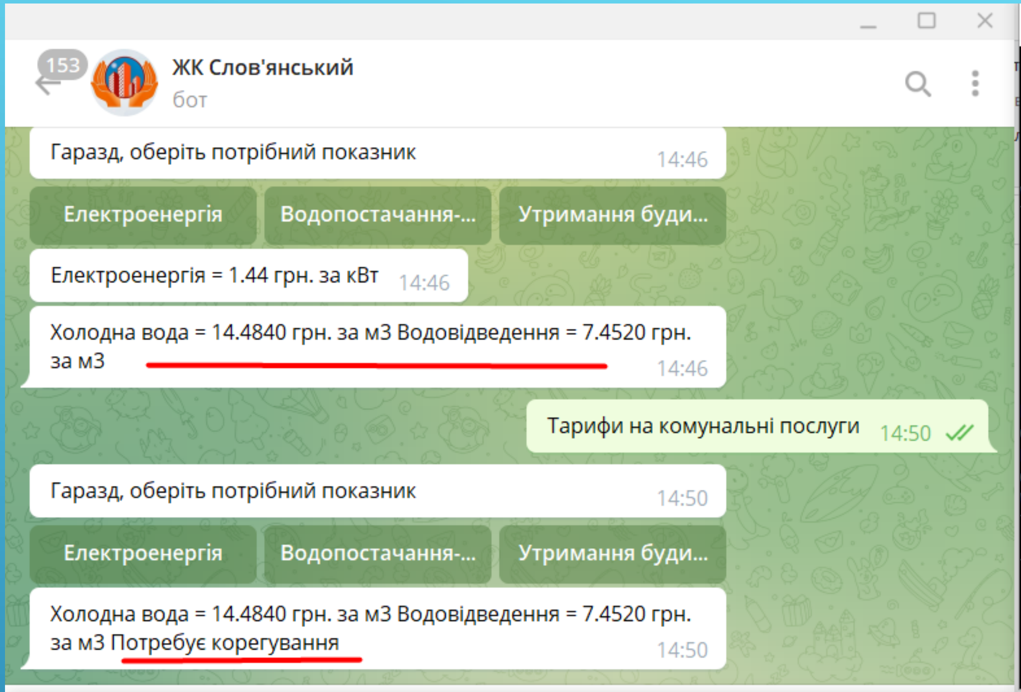

MarchenkoOlexandr Update Jenkinsfile

1 contributor

48 lines (46 sloc) | 1.98 KB

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                echo "------------------------------Build_Started------------------------------"
                        sh 'ls -la'
                        sh 'echo "Build by Jenkins Build# $BUILD_ID" >> version.txt'
                        sh 'cat version.txt'
                        sh 'tar --totals -cvf myproject.tar config.py osbbslovjanskijbot.py version.txt'
                        sh 'tar -tf myproject.tar'
                echo "------------------------------Build Finished-----------------------------"
            }
        }
        stage('Test') {
            steps {
                echo "------------------------------Test Started-------------------------------"
                                sh '''

                                '''
                echo "------------------------------Test Finished------------------------------"
            }
        }
        stage('Artefact') {
            steps {
                echo "------------------Creation artefact Started------------------"
```

ADD PIPELINE AND JENKINS FILE ON GITHUB. ADD NOTIFICATION ON TELEGRAM CHAT.



Гаразд, оберіть потрібний показник 14:46

Електроенергія | Водопостачання-... | Утримання буди...

Електроенергія = 1.44 грн. за кВт 14:46

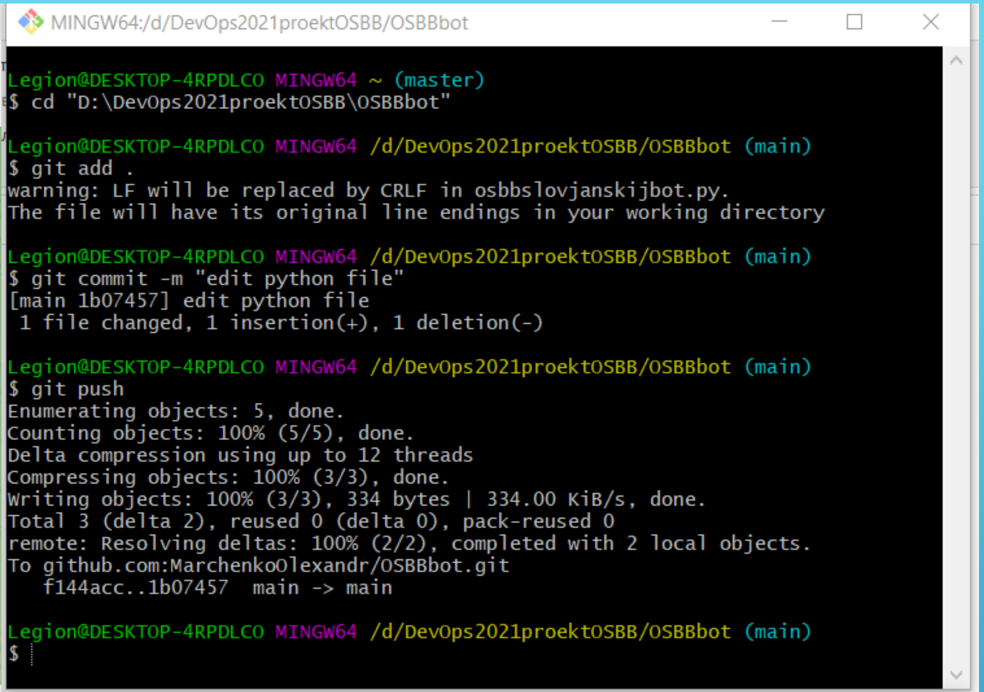
Холодна вода = 14.4840 грн. за м3 Водовідведення = 7.4520 грн. за м3 14:46

Тарифи на комунальні послуги 14:50

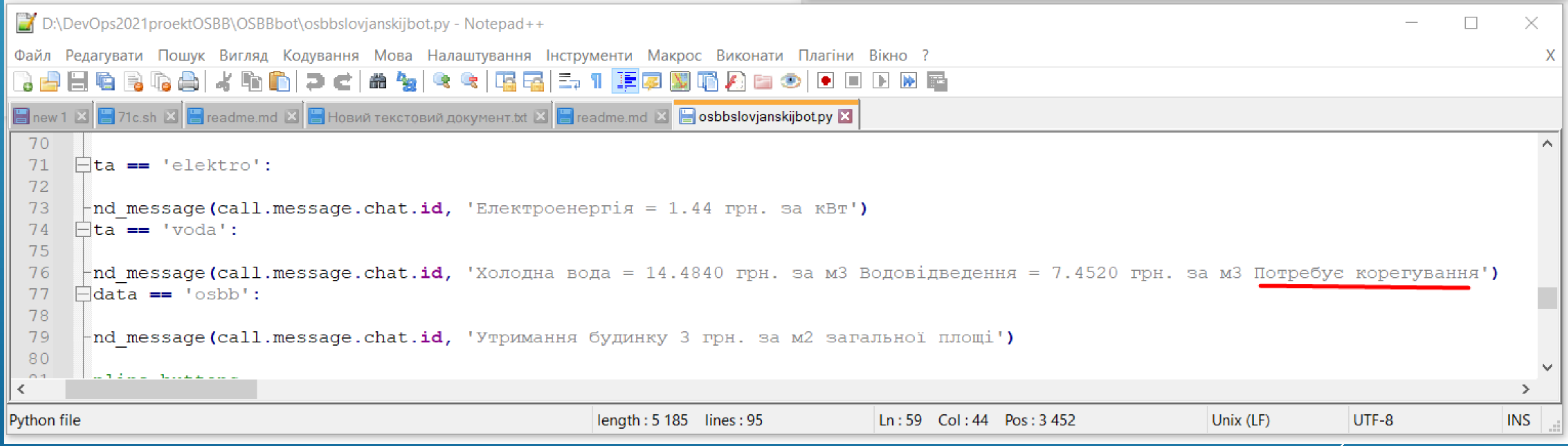
Гаразд, оберіть потрібний показник 14:50

Електроенергія | Водопостачання-... | Утримання буди...

Холодна вода = 14.4840 грн. за м3 Водовідведення = 7.4520 грн. за м3 Потребує корегування 14:50



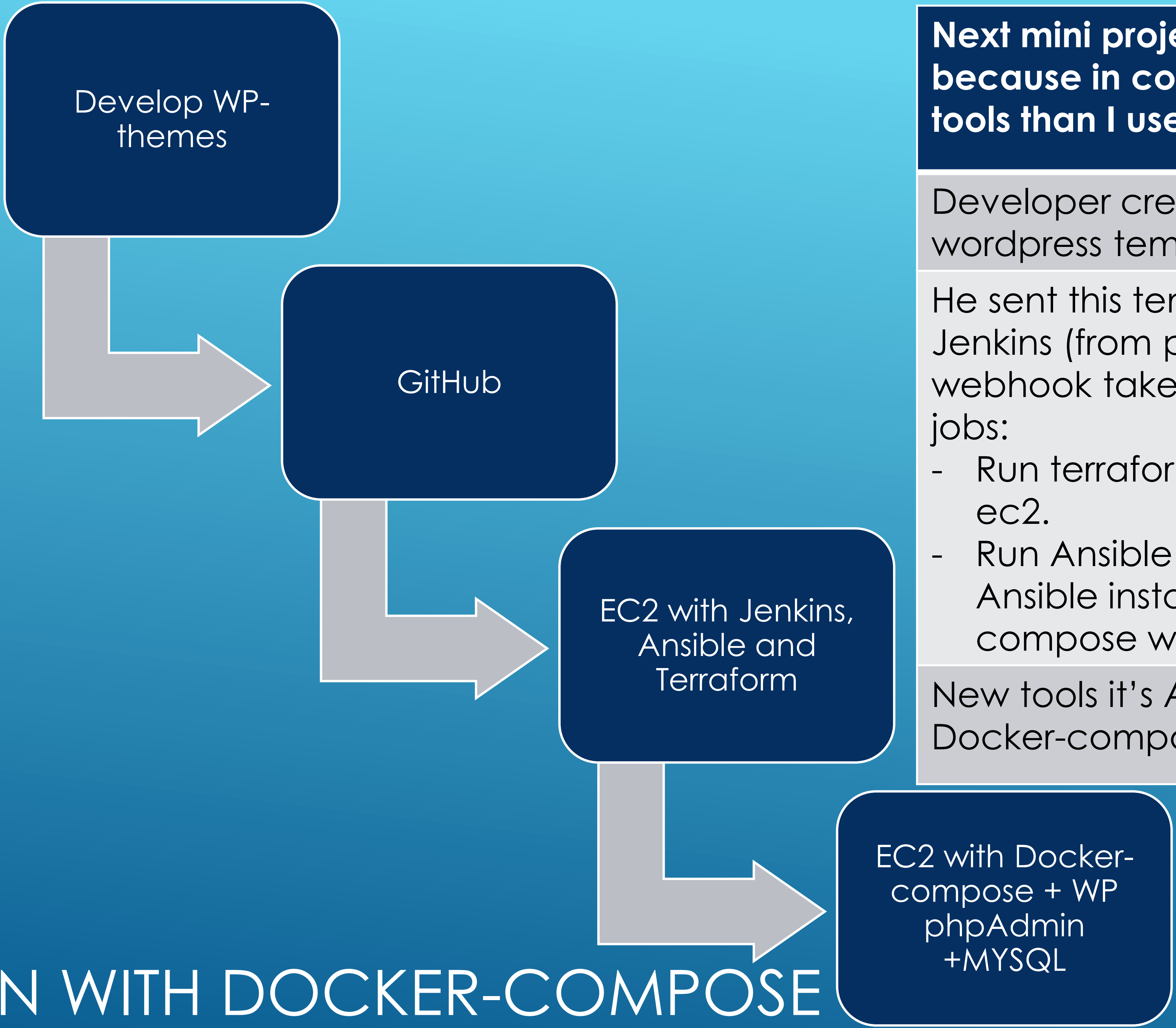
```
Legion@DESKTOP-4RPDLCO MINGW64 ~ (master)
$ cd "D:\DevOps2021proektOSBB\OSBBbot"

Legion@DESKTOP-4RPDLCO MINGW64 /d/DevOps2021proektOSBB/OSBBbot (main)
$ git add .
warning: LF will be replaced by CRLF in osbbslovjanskijbot.py.
The file will have its original line endings in your working directory

Legion@DESKTOP-4RPDLCO MINGW64 /d/DevOps2021proektOSBB/OSBBbot (main)
$ git commit -m "edit python file"
[main 1b07457] edit python file
 1 file changed, 1 insertion(+), 1 deletion(-)

Legion@DESKTOP-4RPDLCO MINGW64 /d/DevOps2021proektOSBB/OSBBbot (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 334 bytes | 334.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:MarchenkoOlexandr/OSBBbot.git
   f144acc..1b07457  main -> main

Legion@DESKTOP-4RPDLCO MINGW64 /d/DevOps2021proektOSBB/OSBBbot (main)
$
```


D:\DevOps2021proektOSBB\OSBBbot\osbbslovjanskijbot.py - Notepad++
Файл Редагувати Пошук Вигляд Кодування Мова Налаштування Інструменти Макрос Виконати Плагіни Вікно ?
new 1
71c.sh
readme.md
Новий текстовий документ.txt
readme.md
osbbslovjanskijbot.py

```
ta == 'elektro':

nd_message(call.message.chat.id, 'Електроенергія = 1.44 грн. за кВт')
ta == 'voda':

nd_message(call.message.chat.id, 'Холодна вода = 14.4840 грн. за м3 Водовідведення = 7.4520 грн. за м3 Потребує корегування')
data == 'osbb':

nd_message(call.message.chat.id, 'Утримання будинку 3 грн. за м2 загальної площі')
```

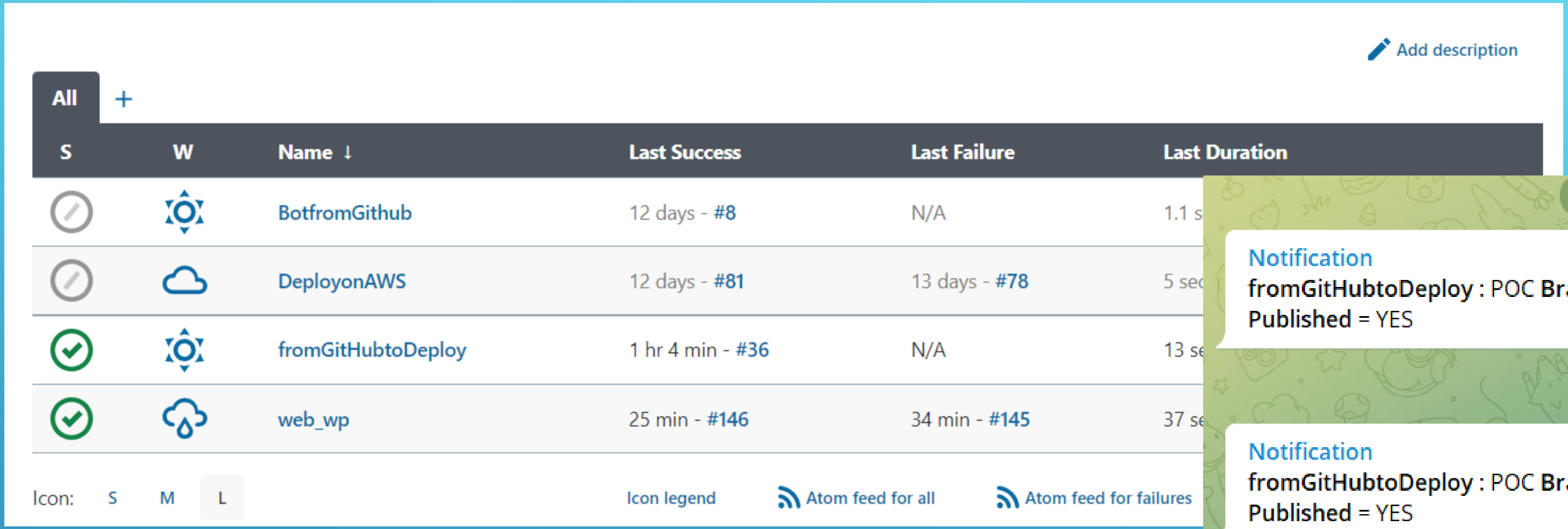
Python file   length : 5 185   lines : 95   Ln : 59   Col : 44   Pos : 3 452   Unix (LF)   UTF-8   INS

DEVELOPER CHANGES PYTHON SCRYPT. RESULT


Develop WP-themes
GitHub
EC2 with Jenkins, Ansible and Terraform
EC2 with Docker-compose + WP phpAdmin +MYSQL

Next mini project. I did this project because in course we learned more tools than I used in previous project.
Developer create and then edit wordpress template
He sent this template on Github. Jenkins (from previous project) with webhook take Jenkinsfile and make his jobs: - Run terraform. Terraform create new ec2. - Run Ansible. Ansible configure ec2. Ansible install and run docker-compose with wordpress.
New tools it's Ansible and Docker, Docker-compose.

WP RUN WITH DOCKER-COMPOSE

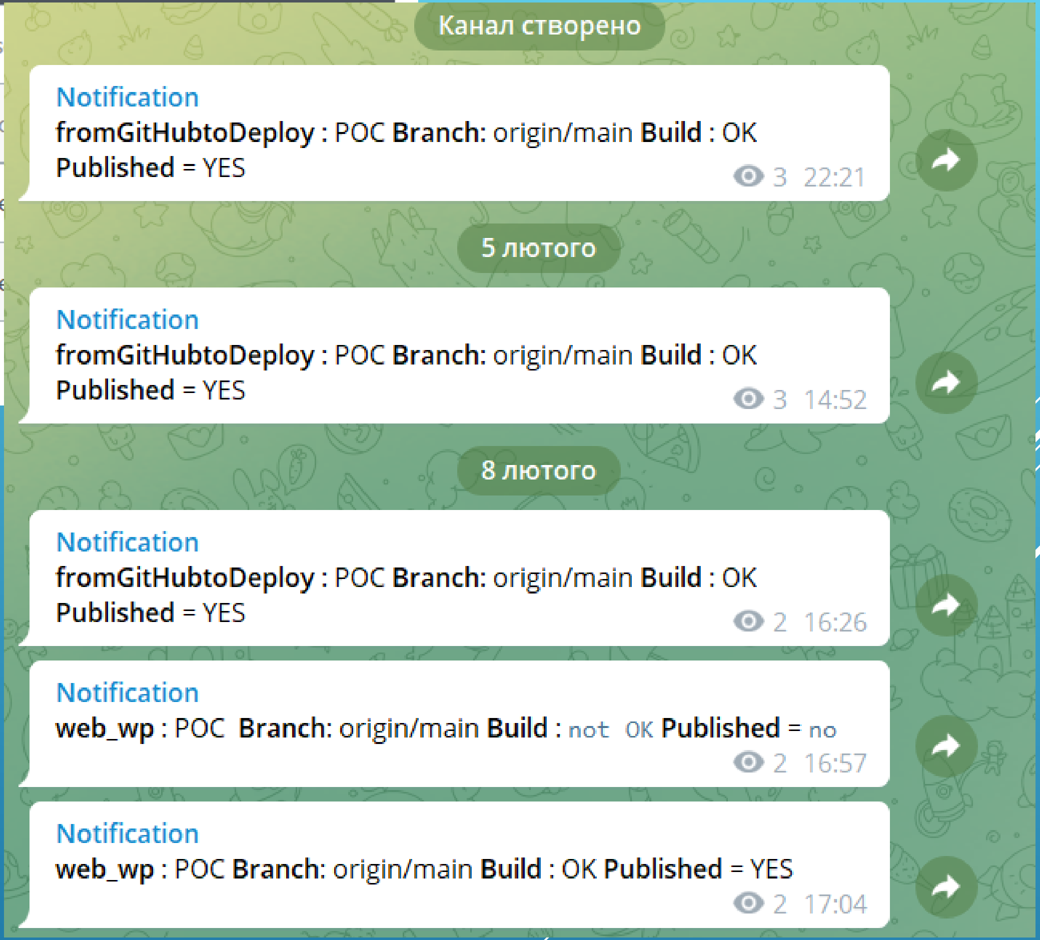


Add description

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration
		BotfromGithub	12 days - #8	N/A	1.1 s
		DeployonAWS	12 days - #81	13 days - #78	5 sec
✓		fromGitHubtoDeploy	1 hr 4 min - #36	N/A	13 se
✓		web_wp	25 min - #146	34 min - #145	37 se

Icon: S M L   Icon legend   Atom feed for all   Atom feed for failures



Канал створено

Notification
fromGitHubtoDeploy : POC **Branch:** origin/main **Build** : OK **Published** = YES
3 22:21

5 лютого

Notification
fromGitHubtoDeploy : POC **Branch:** origin/main **Build** : OK **Published** = YES
3 14:52

8 лютого

Notification
fromGitHubtoDeploy : POC **Branch:** origin/main **Build** : OK **Published** = YES
2 16:26

Notification
web_wp : POC **Branch:** origin/main **Build** : `not OK` **Published** = `no`
2 16:57

Notification
web_wp : POC **Branch:** origin/main **Build** : OK **Published** = YES
2 17:04

NOTIFICATION AND PIPELINE WITH JENKINS FILE FROM GITHUB. TERRAFORM STATE ON AWS S3