Ministry of Higher Education & scientific research

- Horizon School of Digital Technologies -

Department: Software Engineering

Internship Report

Host organisation: **SOFTIFI**

Produced by: **Hamza Lazigheb**

Supervised by : **Mr Achraf Msaddak**

Period :

From August 1st to August 31st

Acknowledgments

Before I commence drafting my report, I wish to extend my heartfelt gratitude to the individuals who supported and encouraged me throughout the execution of this project and the successful completion of my internship.

First and foremost, I would like to express my appreciation to the entire staff of **SOFTIFI** company, with special mention to my supervisor, **Mr. Achraf Msaddak**, for his guidance, assistance, and the quality of mentorship he provided me.

"I am also indebted to the academic school at **Horizon** for equipping me with the knowledge and skills that formed the foundation of this endeavor."

"I would like to thank **Mr. Ridwan Al-Rabee** for giving me the opportunity to enter the professional life."

Table of Contents

List of Figures

Introduction

- In this report, we will delve into the narrative of this enriching experience. By exploring the challenges faced, the creative solutions developed, and the overall impact of the DevOps approach in the professional realm, we will unveil how this immersion goes far beyond a mere internship. It has bolstered my understanding of software development while contributing to my readiness for a dynamic and fulfilling academic and professional career.

- At the heart of this journey lies a significant achievement that marked a pivotal milestone: the successful implementation of continuous integration via Jenkins. This accomplishment, orchestrated with tools such as Jenkins, virtual machines, Bitbucket, and Ngrok, embodied the realization of fundamental DevOps principles. It not only automated and streamlined our development and deployment processes but also fostered seamless collaboration among our development and operations teams. This experience served as a true showcase of the effectiveness of DevOps practices in a tangible professional context, thereby solidifying my expertise in automated deployment and efficient software lifecycle management.

I - Presentation and Activities of SOFTIFI Company :

- SOFTIFI is a comprehensive IT services company that offers high-quality digital solutions. Specializing in web and mobile application development, the company is committed to delivering innovative ideas and ensuring client satisfaction. The core philosophy of SOFTIFI revolves around enhancing these ideas through practical and digital solutions, all while respecting the resources of its clients.

- A notable focus of SOFTIFI lies in providing professional Enterprise Resource Planning (ERP) solutions. The company's expertise and efforts are strategically invested in the development of ERP solutions, particularly utilizing the Odoo platform. This dedication to ERP solutions showcases SOFTIFI's commitment to enhancing its clients' business operations and processes.

Figure 1: SOFTIFI Logo

The headquarters of SOFTIFI is located at Parc d'Activités Economiques Zarzis Smart Center.
Contact Details for SOFTIFI: Phone: +216 21 787 771 Email Address: info@softifi.com

Figure 2: SOFTIFI Headquarters

1. Products and Services Offered:

- Softifi empowers its clients to transform their creative ideas into practical digital solutions. The company offers tailored web and mobile solutions, along with comprehensive electronic marketing services to support its clients' digital transformation journey. The primary objective of SOFTIFI is to provide a comprehensive suite of IT solutions that cater to its clients' needs. Its clientele ranges from businesses of varying sizes, and it delivers personalized services aligned with each company's objectives and specifications.

- To achieve optimal outcomes, SOFTIFI is dedicated to implementing Odoo ERP, CRM, E-commerce, web and mobile development, outsourcing, and digital marketing. Moreover, the company offers customized solutions through the PEGA platform, which harnesses artificial intelligence to cover all technical and business aspects. This approach enables SOFTIFI to effectively address a wide array of technical and business challenges faced by its clients.

II. Presentation of the Accomplished Work:

III.1. Subject Description:

Utilizing a virtual machine (VM) alongside Bitbucket and Ngrok, the continuous integration process is orchestrated via Jenkins. The VM hosts Jenkins for automated build, test, and deployment workflows. Bitbucket serves as the version control system, while Ngrok facilitates secure tunneling for external access to the VM-hosted Jenkins instance, enabling efficient collaboration and seamless integration of code changes.

II.2. Task Description:

II.2.1 Virtualization

Virtualization, like using VirtualBox, creates virtual machines on one physical computer to run multiple OSes and software independently. It optimizes resource use and aids testing.

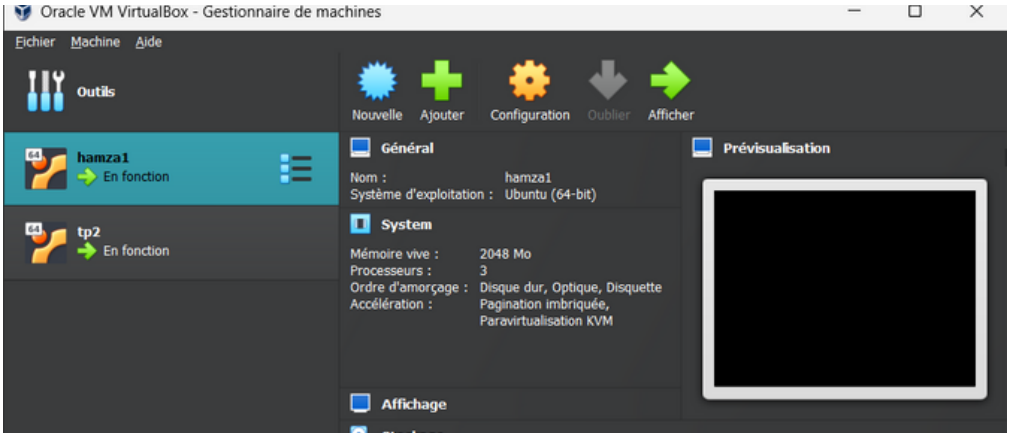


figure 3 : VM

Benefits: Efficient resource utilization, cost savings, isolation of environments, easy migration, and scalability.

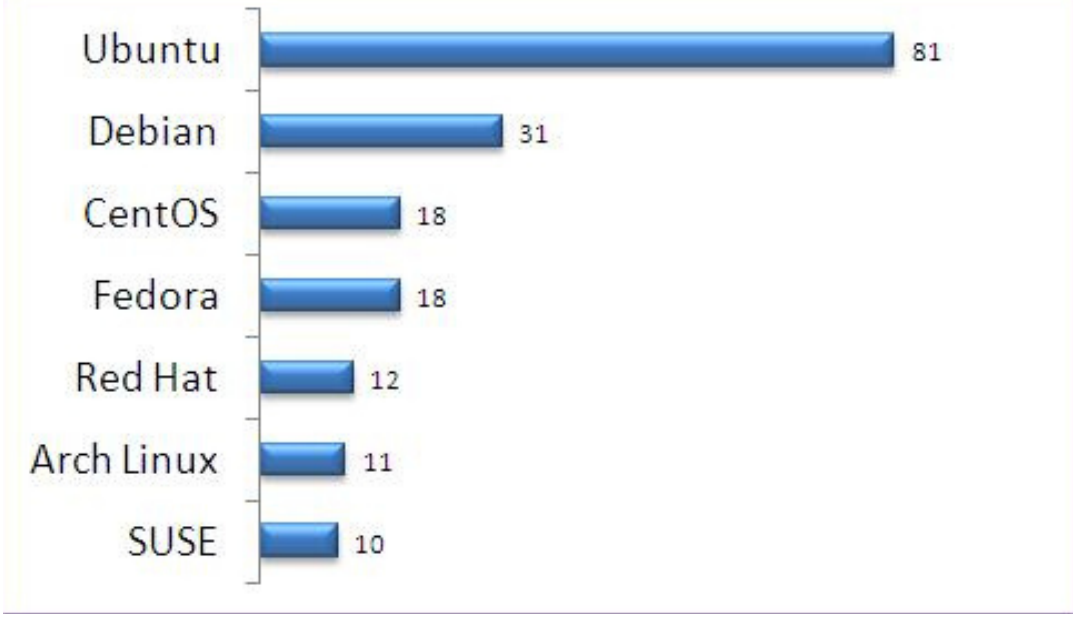


figure 4 : Top 5 Linux Os Distrubtion

II.2.2 SSH connexion

1. What is SSH:

SSH (Secure Shell) is a cryptographic network protocol used to establish secure and encrypted connections between two systems over an unsecured network. It provides a secure way to access and manage remote systems.

2. Key Components of SSH:

SSH involves various components, including:

Public and Private Keys: Used for authentication and encryption.
Host Keys: Used to verify the authenticity of a host.
Authentication Methods: Password-based, key-based, and more.

ssh-keygen -t rsa -b 4096 -C "hamzalazigheb@gmail.com"
cat /.ssh/ida_rsa.pub

```
mza1@hamza1:~/.ssh$ cat id_rsa.pub
h-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDWvc5/itw7q4B6reYkCB+YAB9TxKZYBn+m6ZndV/n2Ulg3+8
eE5RcT3K/3+PWJGl68rGMc4qXOW9HZDS2khfh9vmytUuPqXUM9Zd1o5AlBS4b3i51z5HyyzcO/TntaXmR9GV
XLo8zI/fqCjuhfxNdfnKPtqBAkBKGRNDweW49ZA3z1dXcw7hCJTN9pXfp58ypAZyF/3OzvZ0KArPfT34oh7f
ynUx94cd6AXGHQrDSi7LlCgWstI6eXmCiq4KjBpVSRBW8vvKhCZGBVtjBstkISjLFSReWk719OQtMMOSoAON
oo/QpW/+DSXidyTnDBF/pQOaJM5tEcwYPyvbCc6rO1kRuiz6zuIREgoZQrr5DHKAaQ4/obks6ngY+qKZZBwW
B1NIpkhpPflNSyFbkwciYTmdfbBKEBySxmXnqChyv575L35QuL3T7YEywefQ== hamzalazigheb@gmail.c
mza1@hamza1:~/.ssh$
```

figure 5 : Public Key

```
amza1@hamza1:~/.ssh$ cat id_rsa
----BEGIN OPENSSH PRIVATE KEY-----
3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAdzc2gtcn
hAAAAAwEAAQAAAgEA1r3Of4rcO6uAeq3mJAgfmAAfU8SmWAZ/pumZ3Vf59lJYN/vNg+8q
LWFMJwxgbVvgwn4XIRT76fgfh5LL+FQpTEwZIZRgn4nhOUXE9yv9/j1iRpevKxjHOKlzl
R2Q0tpIX4fb5srVLj6l1DPWXdaOQJQUuG94udc+R8ss3Dv057Wl5kfRlVARK31PItejQp
YXpRfF0xElvhNsaT/VU3OmuQBDH04mVoyI1y6PMyP36go7oX8TXX5yj7agQJAShkTQ8Hl
PWQN89XV3MO4QiUzfaV36efMqQGchf9zs72dCgKz309+KIe375k15t1JmwQgU0WQm0L/J
SWbTqtSejw5JhrwMxwoNLfsuUYsp1MfeHHegFxh0Kw0ouy5QoFrLSOnl5goquCowaVUkQ
vL7yoQmRgVbYwbLZCEoyxUkXlpO9fTkLTDDkqADjVWyjeBmBBnbtk2x1DKIG0F7oFh/N0
QbwPPJnYuFkpQ4E1bHaKP0KVv/g0l4nck5wwRf6UDmiTObRHMGD8r2wnOqztZEbos+s7i
RIKGUK6+QxygGkOP6G5LOp4GPqimWQcFnYibcraKS1taUzmkAR+8UoGqEC5mLGMZ/jK1s
jtX35BmdeOAdTSKZIaT35TUshW5MHImE5nX2wShAcksZl56gocr+e+S9+ULi90+2BMsHn
AAAdQ+IYXSPiGF0gAAAAHc3NoLXJzYQAAAgEA1r3Of4rcO6uAeq3mJAgfmAAfU8SmWAZ/
umZ3Vf59lJYN/vNg+8qFLWFMJwxgbVvgwn4XIRT76fgfh5LL+FQpTEwZIZRgn4nhOUXE9
v9/j1iRpevKxjHOKlzlvR2Q0tpIX4fb5srVLj6l1DPWXdaOQJQUuG94udc+R8ss3Dv057
l5kfRlVARK31PItejQpvYXpRfF0xElvhNsaT/VU3OmuQBDH04mVoyI1y6PMyP36go7oX8
XX5yj7agQJAShkTQ8HluPWQN89XV3MO4QiUzfaV36efMqQGchf9zs72dCgKz309+KIe37
k15t1JmwQgU0WQm0L/J9SWbTqtSejw5JhrwMxwoNLfsuUYsp1MfeHHegFxh0Kw0ouy5Qo
rLSOnl5goquCowaVUkQVvL7yoQmRgVbYwbLZCEoyxUkXlpO9fTkLTDDkqADjVWyjeBmBB
btk2x1DKIG0F7oFh/N00QbwPPJnYuFkpQ4E1bHaKP0KVv/g0l4nck5wwRf6UDmiTObRHM
D8r2wnOqztZEbos+s7iERIKGUK6+QxygGkOP6G5LOp4GPqimWQcFnYibcraKS1taUzmkA
+8UoGqEC5mLGMZ/jK1s7jtX35BmdeOAdTSKZIaT35TUshW5MHImE5nX2wShAcksZl56go
r+e+S9+ULi90+2BMsHn0AAAADAQABAAACADfGaOIbsQQWkXJLxKDLM1mJfpmFVCc82N8D
aFJ/nSKArvXvRNthSKH2yY7C2jDNUl1x3pQLHTsx5Q6ThELlqSxV/5kHLumBKGx1wKjqV
g4vVPUhfzNnMEM0ib9K9JivdWjKosUZHbG7N9sb86u/1IbVb2RQH601SMng7a/uKrgcKB
dyoPbnfoLvAgytgbQ76vxIZmth4smxxdC4cbIRMdaPSN/SroaATfKF90+zIes56S8TEHa
```

figure 6 : private Key

Command to generate SSH

1. cat ~/.ssh/id_rsa.pub
2. mkdir -p ~/.ssh
3. nano ~/.ssh/authorized_keys
4. chmod 700 ~/.ssh chmod 600 ~/.ssh/authorized_keys

NB

Public Key:

Component of asymmetric cryptography.
Shared openly.
Used for encryption and verifying digital signatures.
Freely distributed and usable by anyone.
Derived from private key, computationally hard to reverse-engineer.
Sends encrypted messages to owner of the matching private key.
Enables secure communication without sharing private key.

Private Key:

Component of asymmetric cryptography.
Kept secret and secure.
Decrypts messages encrypted with corresponding public key.
Digitally signs messages for authenticity and non-repudiation.
Ownership and control critical for security.
Never shared.
Typically generated in a pair with corresponding public key.
In summary, public and private keys are asymmetric cryptography key pairs. Public key encrypts and verifies, private key decrypts and signs, enabling secure communication and verification.

3. SSH Client and Server:

An SSH client initiates a connection request, while an SSH server accepts and processes incoming connection requests.

4. SSH Key Generation:

Generate SSH key pairs using the ssh-keygen command. The public key is placed on the server, while the private key is kept secure on your local machine.

5. SSH Authentication Methods:

Password-based: Requires entering a password.
Key-based: Uses public and private keys for authentication, providing enhanced security.

6. Passwordless SSH Login:

Setting up passwordless login involves generating an SSH key pair, adding the public key to the remote server's ~/.ssh/authorized_keys file, and using the private key for authentication.

7. SSH Configuration File:

The SSH configuration file (~/.ssh/config) allows customizing connection settings, including aliases, port numbers, and more.

8. SSH Tunneling (Port Forwarding):

SSH tunneling allows forwarding network connections through an encrypted SSH connection, useful for accessing remote services securely.

II.2.3 Scripting shell

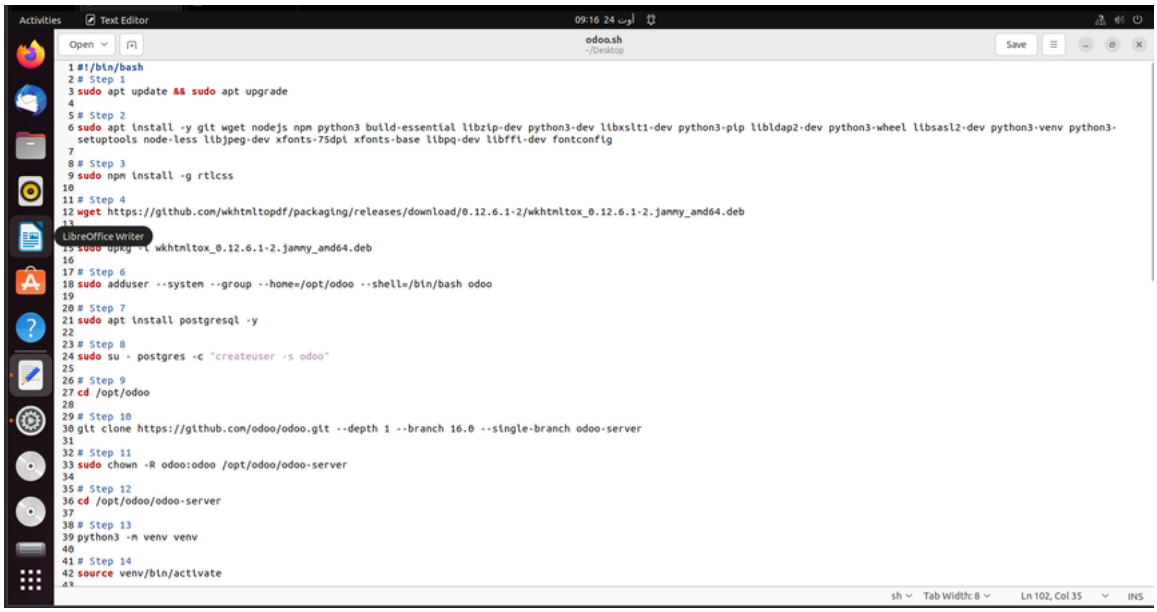
a-Definition

Shell scripting utilizes languages like Bash to automate tasks via system commands in the command line, simplifying actions such as file management and system configuration. While straightforward, complex tasks demand command comprehension and basic programming skills, making it an efficient way to automate OS interactions through text-based commands.

b- install odoo via shell script

Installing Odoo via shell script automates setup with system updates, dependency installations (Python, PostgreSQL), and Odoo configuration. Tailoring the script is essential for environment-specific adjustments, ensuring secure deployment through official guidelines.

```
# This is a basic shell script example that demonstrates installation Odoo
```




```
#!/bin/bash
# Step 1
sudo apt update && sudo apt upgrade

# Step 2
sudo apt install -y git wget nodejs npm python3 build-essential libzip-dev python3-dev libxslt1-dev python3-pip libldap2-dev python3-wheel libsasl2-dev python3-venv python3-
  setuptools node-less libjpeg-dev xfonts-75dpi xfonts-base libpq-dev libffi-dev fontconfig

# Step 3
sudo npm install -g rtlcss

# Step 4
wget https://github.com/wkhtmltopdf/packaging/releases/download/0.12.6.1-2/wkhtmltox_0.12.6.1-2.jammy_amd64.deb

sudo dpkg -i wkhtmltox_0.12.6.1-2.jammy_amd64.deb

# Step 6
sudo adduser --system --group --home=/opt/odoo --shell=/bin/bash odoo

# Step 7
sudo apt install postgresql -y

# Step 8
sudo su - postgres -c "createuser -s odoo"

# Step 9
cd /opt/odoo

# Step 10
git clone https://github.com/odoo/odoo.git --depth 1 --branch 16.0 --single-branch odoo-server

# Step 11
sudo chown -R odoo:odoo /opt/odoo/odoo-server

# Step 12
cd /opt/odoo/odoo-server

# Step 13
python3 -m venv venv

# Step 14
source venv/bin/activate
```

figure 7 : script odoo instalation

II.2.4 Jenkins configuration

a - Jenkins VM connexion

- **Introduction**:

In the realm of modern software development, the combination of Jenkins, a powerful automation tool, with virtual machines (VMs) offers a gateway to efficient and scalable continuous integration. This report delves into the synergy of Jenkins and VMs, outlining their pivotal role in optimizing development workflows.

figure 8 : Logo Jenkins

- **Setting up Jenkins on Vm ：**

Access VM: Connect to your VM via SSH using a terminal or an SSH client: ssh username@vm_ip.

Update System:

Update package information: sudo apt update

Upgrade installed packages: sudo apt upgrade

Install Java: Jenkins requires Java. Install OpenJDK: sudo apt install openjdk-11-jdk.

Install Jenkins:

Add Jenkins repository key: wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

Add Jenkins repository: sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

Update package list: sudo apt update

Install Jenkins: sudo apt install jenkins

Start Jenkins:

Start service: sudo systemctl start jenkins

Enable auto-start: sudo systemctl enable jenkins

Access Jenkins Web UI:

Open a browser and enter http://<VM_IP>:8080.

Retrieve initial admin password: sudo cat /var/lib/jenkins/secrets/initialAdminPassword.

Follow setup wizard, installing recommended plugins.

Configure Firewall (if needed):

Allow incoming traffic on port 8080: sudo ufw allow 8080

Secure Jenkins:

Change admin password.

Set up user accounts and permissions.

Configure security settings as needed.

Install Plugins:

Access "Manage Jenkins" > "Manage Plugins."

Install necessary plugins for your CI/CD workflow.

Create SSH Keys (Optional):

Generate SSH key pair for authentication: ssh-keygen

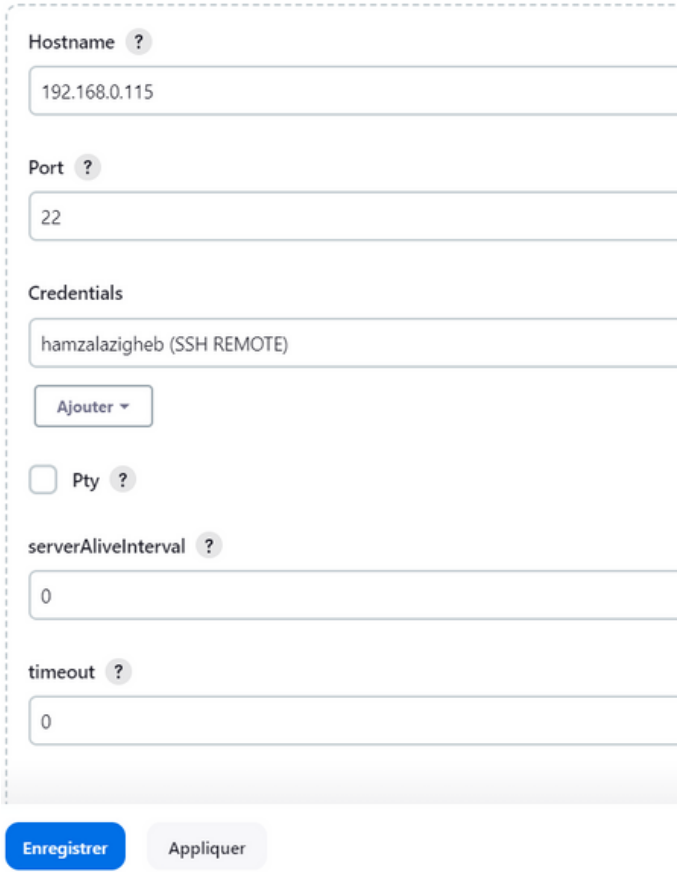


figure 9 : SSH setting

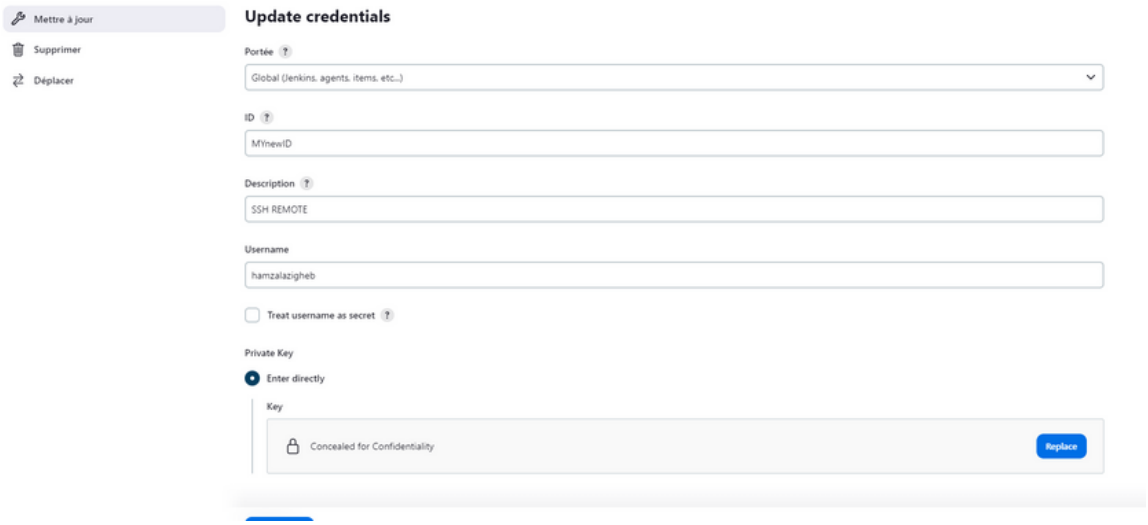


figure 10 : Credentiel setting

b -Jenkins Bitbucket connexion

b.1 introduction

Linking Jenkins with Bitbucket streamlines development through automated CI/CD pipelines, simplifying code integration, testing, and deployment processes.

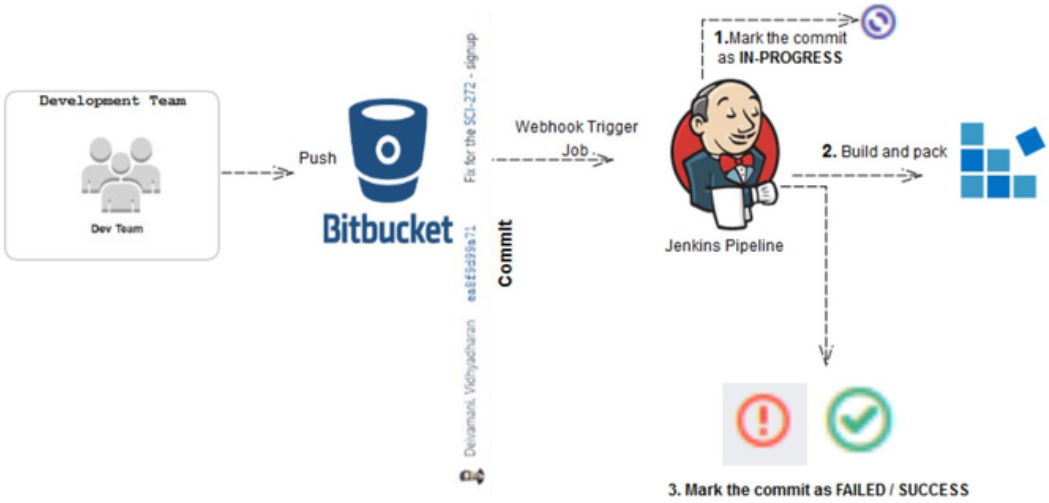


figure 11 : jenkins - Bitbucket

b.2 Setting up

Access Bitbucket:

Log in to your Bitbucket account (https://bitbucket.org).

Create Repository:

Create a new repository or choose an existing one.

Create Webhook in Bitbucket:

In Bitbucket, go to your repository's "Settings" > "Webhooks."

Add a new webhook:

URL: Enter the Jenkins webhook URL (http://jenkins-server:port/bitbucket-hook/).

Secret: Optionally, use a shared secret for added security.

Events: Select the events that should trigger the webhook (e.g., Push, Pull Request).

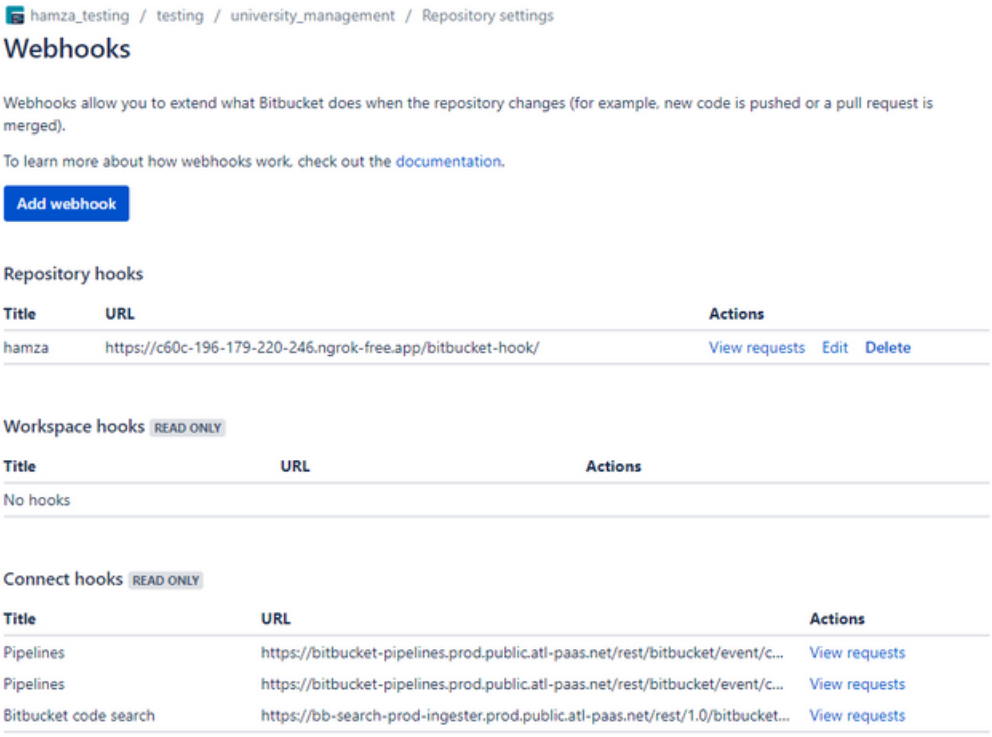


figure 12 : Webhook

II.2.5 Ngrok

Ngrok streamlines web development by securely exposing local servers online for testing and collaboration. In my project, Ngrok was crucial in connecting my VM to the internet, enabling seamless interaction with Bitbucket's automated processes. This integration optimized our CI/CD workflow, highlighting the efficiency of merging tools for a robust software development lifecycle.

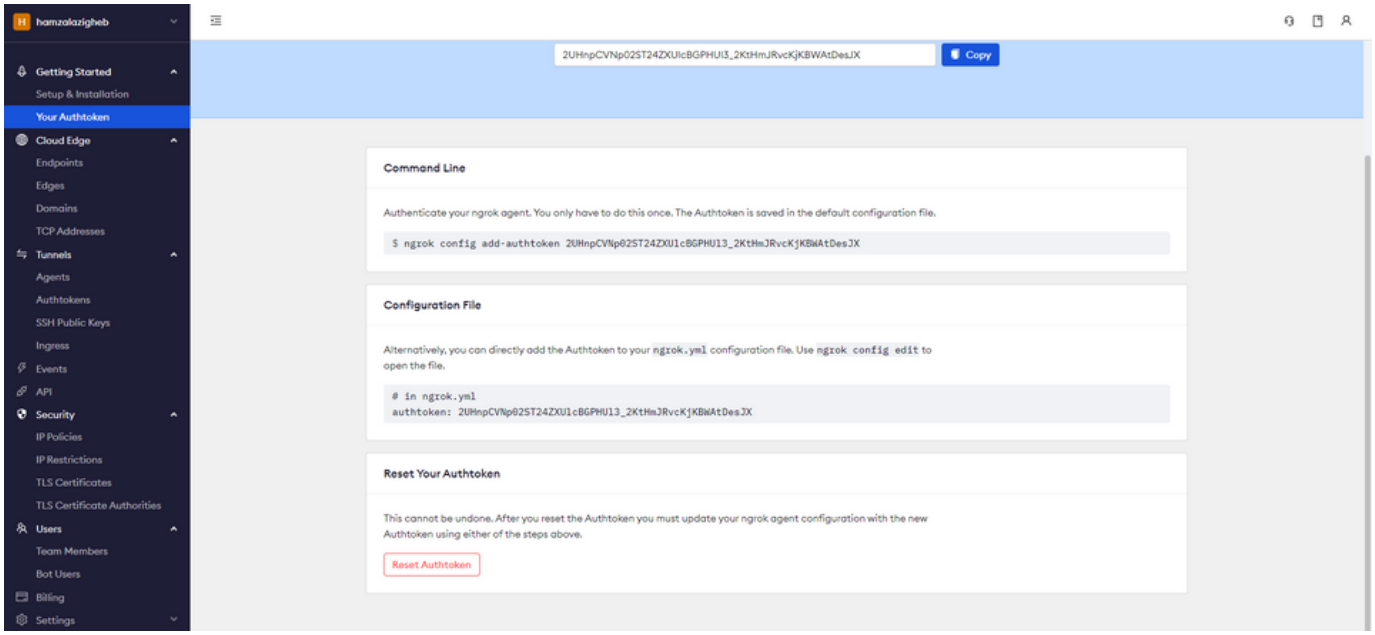


figure 13: inetrface of Ngrok

- **<u>Step By Step : Configuration</u>**

1. Download and Extract:
2. Download and extract ngrok: unzip ngrok.zip.
3. Sign Up and Authenticate:
4. Authenticate your account: ./ngrok authtoken <your_auth_token>.
5. Expose Local Server:
6. Start your local server (e.g., on port 8000).
7. Create a Secure Tunnel:
8. Create a tunnel: ./ngrok http 8000.
9. Access Public URL:
10. Access the generated public URL (e.g., http://<unique_id>.ngrok.io).

II.2.6 Job Jenkins

In my Jenkins job, I leveraged SSH connectivity to seamlessly update custom Odoo addons. This method facilitated efficient remote access for keeping the addons up-to-date and enhancing the Odoo customization process.

figure 14 : job

<u>Setting Up SSH Job in Jenkins:</u>

Create New Job:
Log in to Jenkins and create a new Freestyle project job.
Configure Source Control (Optional):
Set up version control settings if needed.
Build Environment:
Enable "Execute shell" or "Execute Windows batch command" based on your system.
SSH Key Setup:
Ensure Jenkins has access to the SSH key or generate one within Jenkins.
Shell Commands:
Add remote server commands using SSH syntax (e.g., ssh user@server "command").
Credentials:
Add remote server credentials (SSH private key or password).
Save and Build:
Save the configuration and run a build to test the SSH connection.
Console Output:
Review the build's console output for SSH command execution.

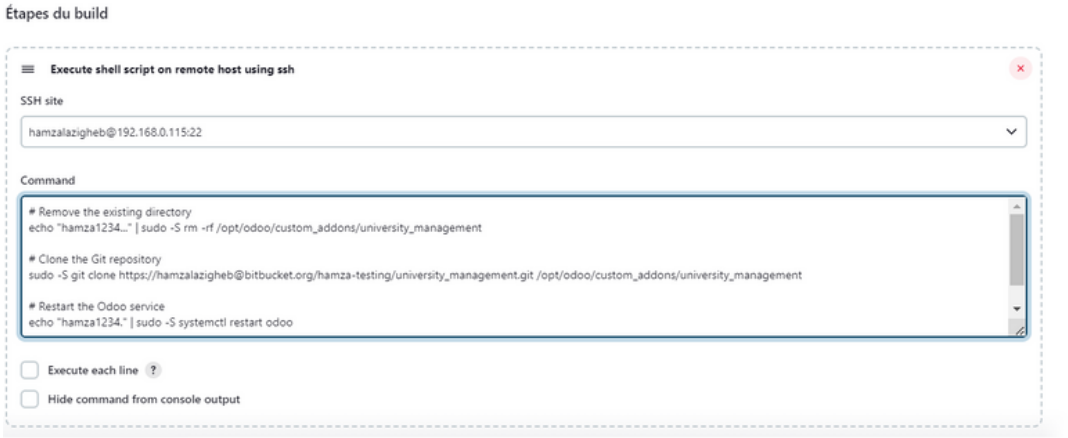


figure 15 : setting up - Job

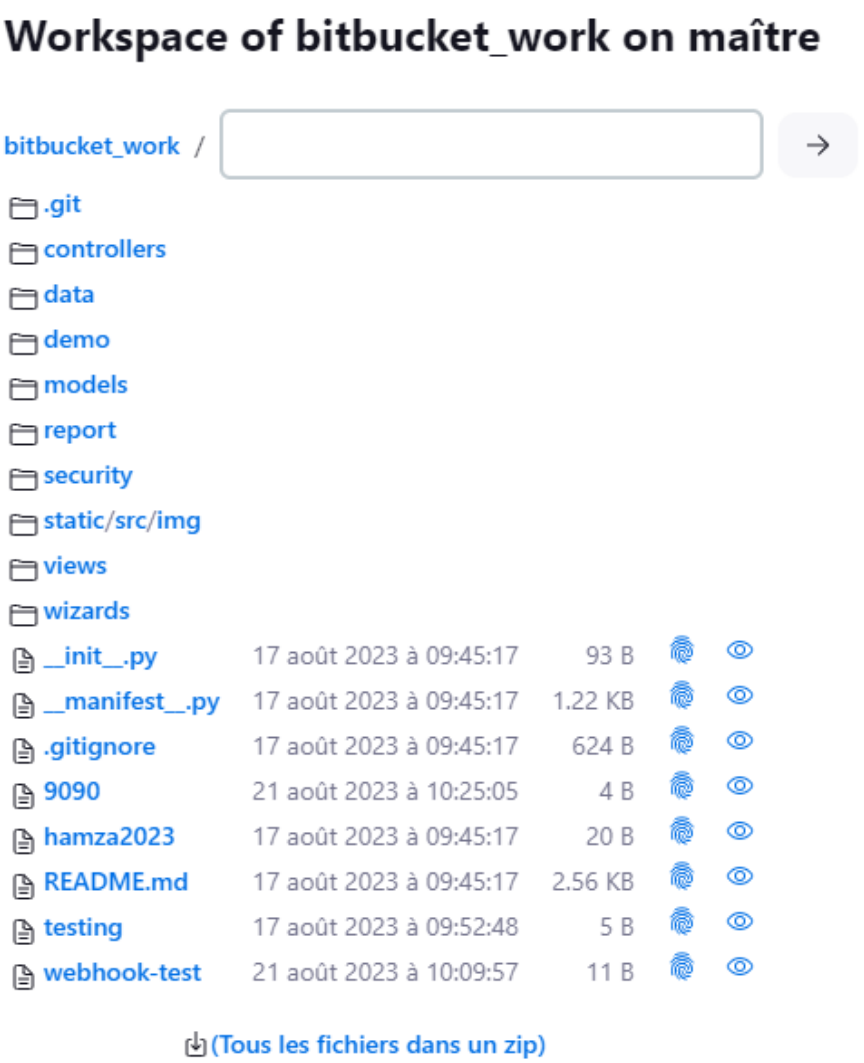


figure 16 : workspace

CONCLUSION

In conclusion, Section II.2's Task Description comprehensively explores technical realms, encompassing Virtualization, SSH Connection, and Scripting Shell. It effectively balances theory and real-world application, addressing Odoo's Shell Script deployment and seamless Jenkins integration with VMs and Bitbucket. This holistic approach offers insights and practical guidance. The inclusion of Ngrok enhances technical depth, culminating in detailed SSH Job directives within Jenkins. This synthesis deepens comprehension, embracing diverse intricacies, highlighting
For my PFE, I'll be focusing on Ansible for deployment and monitoring. This will allow me to delve into efficient deployment strategies and effective software management, aligning with modern DevOps approaches.