

Project Title: *Step-by-Step Guide to Creating a T-Pot Honeypot Using Amazon AWS*

Author: imafidon joseph eromosele

Date: Tuesday 27 may 2025

Table of Contents:

Introduction

- Project purpose
- What is a honeypot?
- Overview of T-Pot and its components
- Why Amazon AWS is used for this setup
- Purpose of the document: to guide users through the installation and deployment process with screenshots

Requirements

- AWS account
- Basic Linux/command line knowledge
- SSH client (e.g., PuTTY or Terminal)
- Recommended instance type (e.g., t3.large or higher)
- Internet connection
- Estimated setup time

Step-by-Step Guide

Launching an EC2 Instance

- Choosing AWS region
- Selecting Ubuntu Server 20.04 LTS
- Choosing instance type
- Setting key pair
- *[Insert screenshot of EC2 setup]*

Configuring Security Groups

- Opening ports: 22 (SSH), 80, 443, 64297, etc.
- *[Insert screenshot of security settings]*

Connecting to EC2 via SSH

- SSH command example and explanation
- *[Insert terminal screenshot]*

Installing Required Dependencies

- Update and install: Git, Docker, Docker Compose
- Commands:

```
bash
CopyEdit
sudo apt update && sudo apt upgrade -y
sudo apt install git docker.io docker-compose -y
```

- *[Insert terminal screenshot]*

Cloning and Installing T-Pot

- Clone GitHub repo:
git clone https://github.com/telekom-security/tpotce
- Navigate to the installer directory:
cd tpotce/iso/installer
- Run installer script:
sudo ./install.sh
- Select installation mode
- *[Insert installation process screenshots]*

Reboot and Initial Login

- Wait for system to reboot
- Login with new credentials
- *[Insert login screen screenshot]*

Accessing the T-Pot Web Interface

- Find public IP of EC2 instance
- Access T-Pot via browser: https://<public-ip>
- Login using web credentials
- *[Insert screenshot of dashboard]*

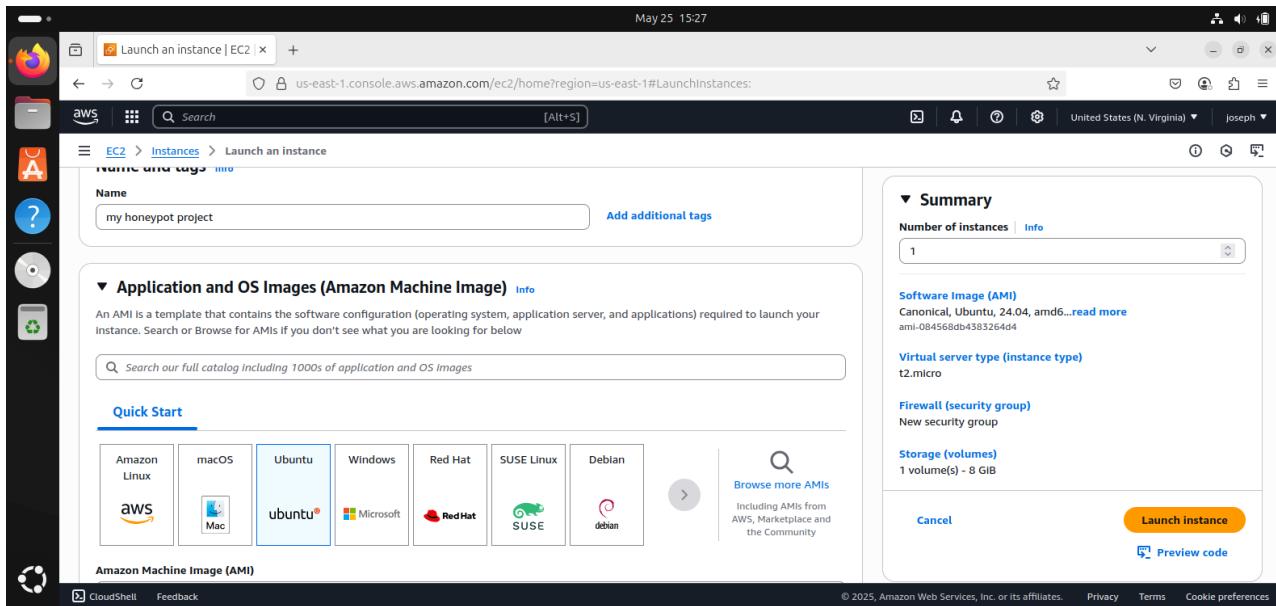
Overview of T-Pot Components

- Kibana: log and traffic visualization
- Suricata: intrusion detection
- Cowrie, Dionaea, etc.
- *[Insert component screenshots]*

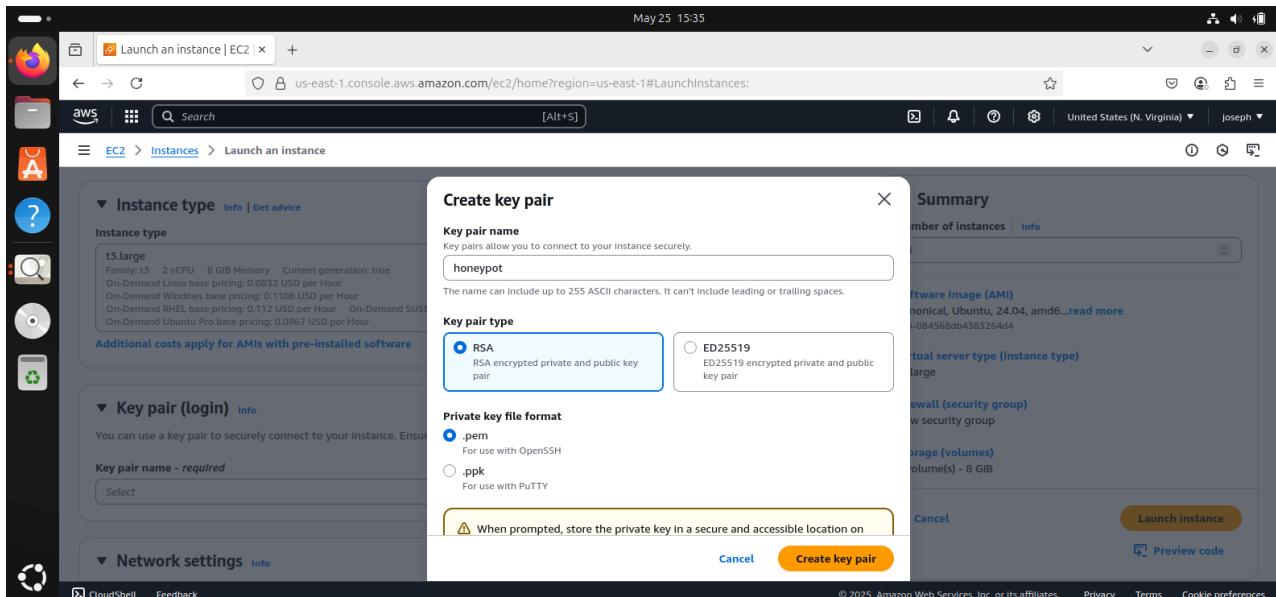
Project purpose:

This outline is structured for easy reading and documentation purposes, ideal for a report, guide, or instructional handout. It includes each major section and what should be covered within it, including screenshots.

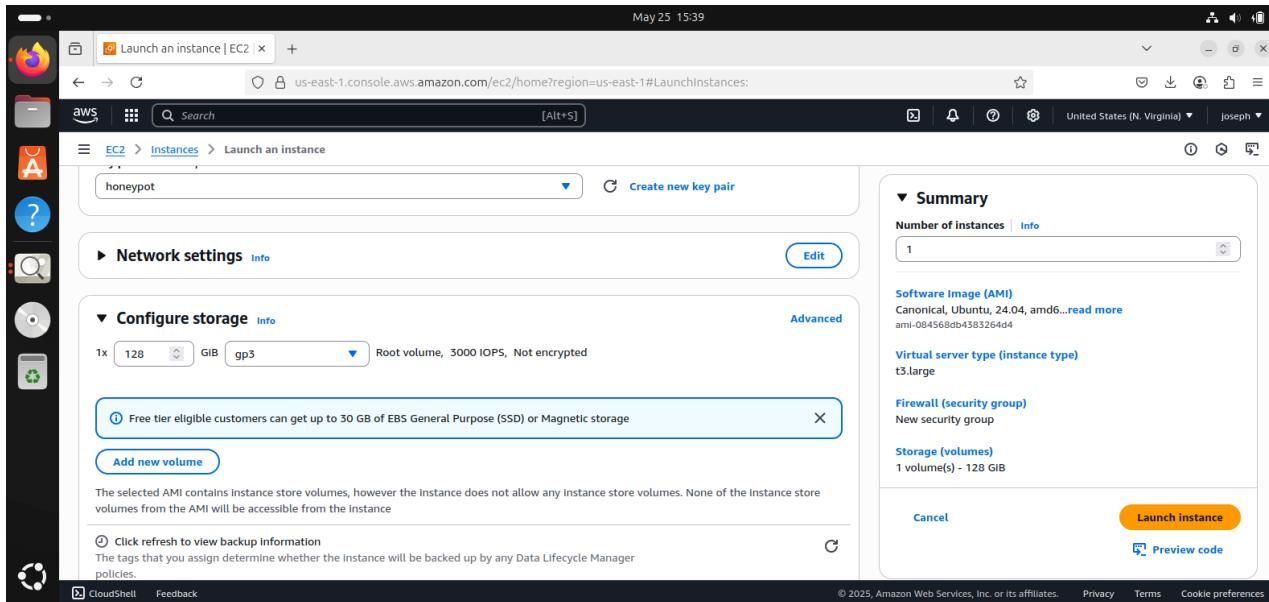
The first step is to have Amazon AWS account, then login and go to EC2 then navigate to instance.



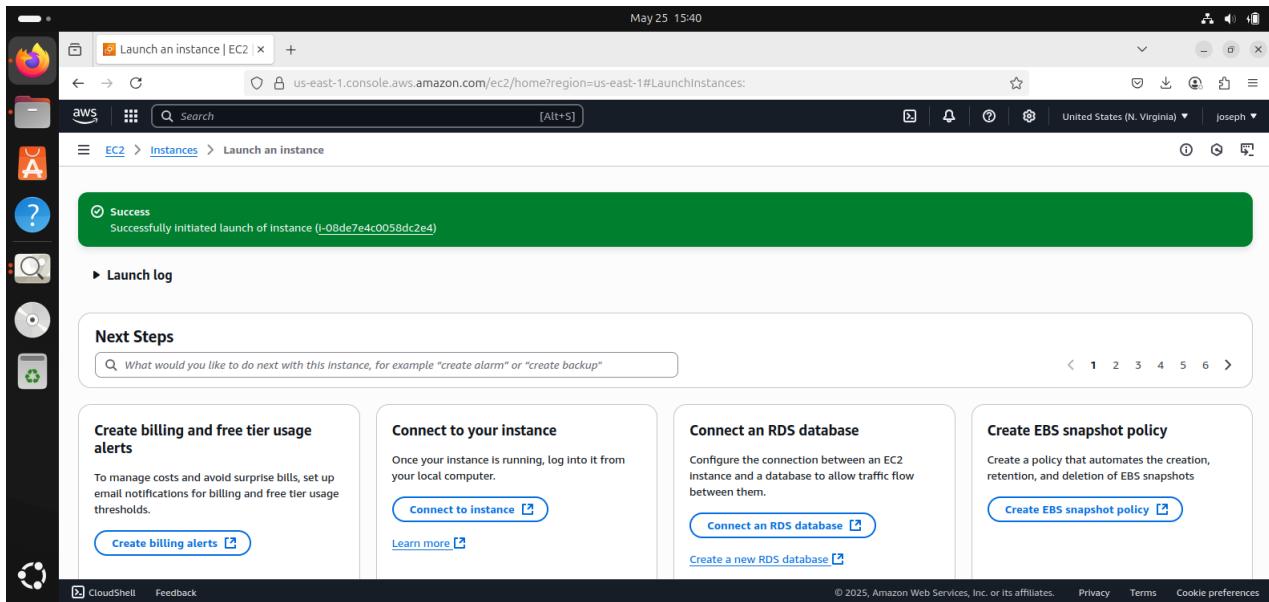
Name your honeypot project, like I did above, then under the quick start option choose Ubuntu.



On the instances type choose t3 large, then create a key pair by choosing your preferred key pair name and leave the key pair type in its default (RSA), private key file format should also be in its default (.pem), and then create key pair.



On configuration storage you can choose a storage size of your choice but for the cost of this project I'm using 128gb, then proceed to launch instance.



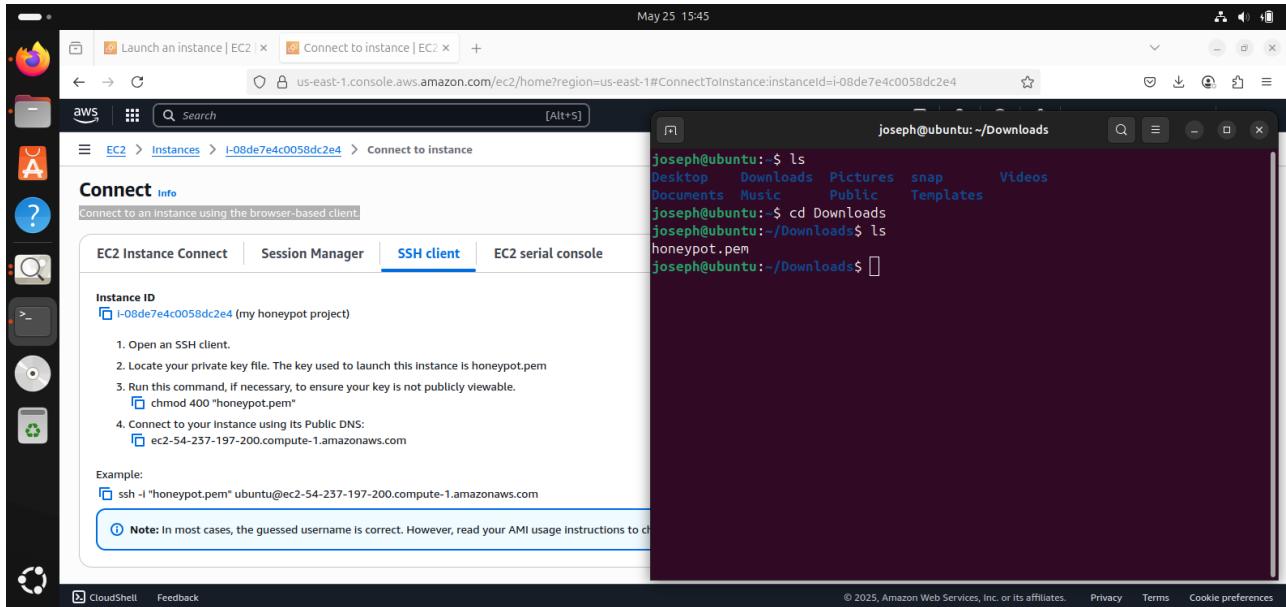
After successfully launching instance, click on the link on the on the instance to take you to the instance dashboard.

The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar and a filter dropdown set to 'All states'. A single instance is listed: 'my honeypot ...' (Instance ID: i-08de7e4c0058dc2e4), which is 'Running' and has an 't3.large' instance type. The status is 'Initializing'. Below the table, a section titled 'i-08de7e4c0058dc2e4 (my honeypot project)' is expanded, showing the 'Details' tab selected. Under 'Instance summary', it shows the Instance ID (i-08de7e4c0058dc2e4), Public IPv4 address (54.237.197.200), Private IPv4 addresses (172.31.28.41), and Public DNS (ec2-54-237-197-200.compute-1.amazonaws.com). There are also tabs for Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

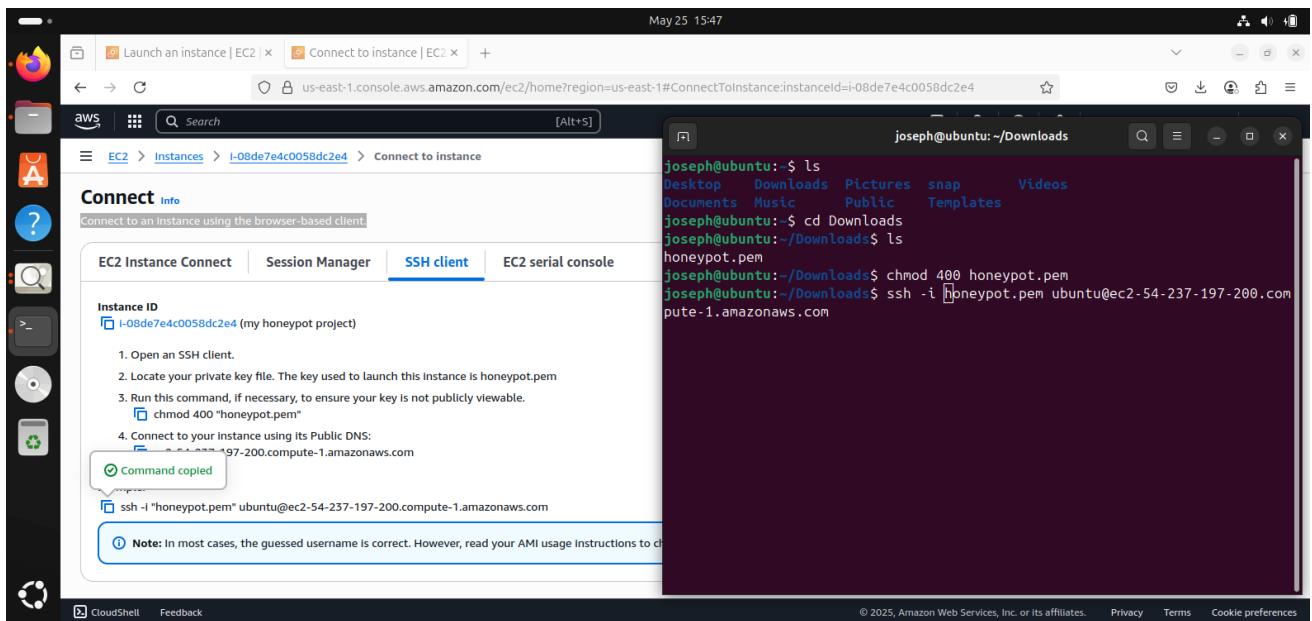
Click on the tick box on the instance and click connect at the top of the instance, you will be directed to a instance connection page.

The screenshot shows the 'Connect' page for the instance i-08de7e4c0058dc2e4. The 'SSH client' tab is selected. It provides instructions for connecting via SSH: '1. Open an SSH client.', '2. Locate your private key file. The key used to launch this instance is honeypot.pem.', '3. Run this command, if necessary, to ensure your key is not publicly viewable.' (with a link to 'chmod 400 "honeypot.pem"'), and '4. Connect to your instance using its Public DNS:' (with a link to 'ec2-54-237-197-200.compute-1.amazonaws.com'). Below these instructions, there's an 'Example:' section with a link to 'ssh -i "honeypot.pem" ubuntu@ec2-54-237-197-200.compute-1.amazonaws.com' and a note: 'Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.'

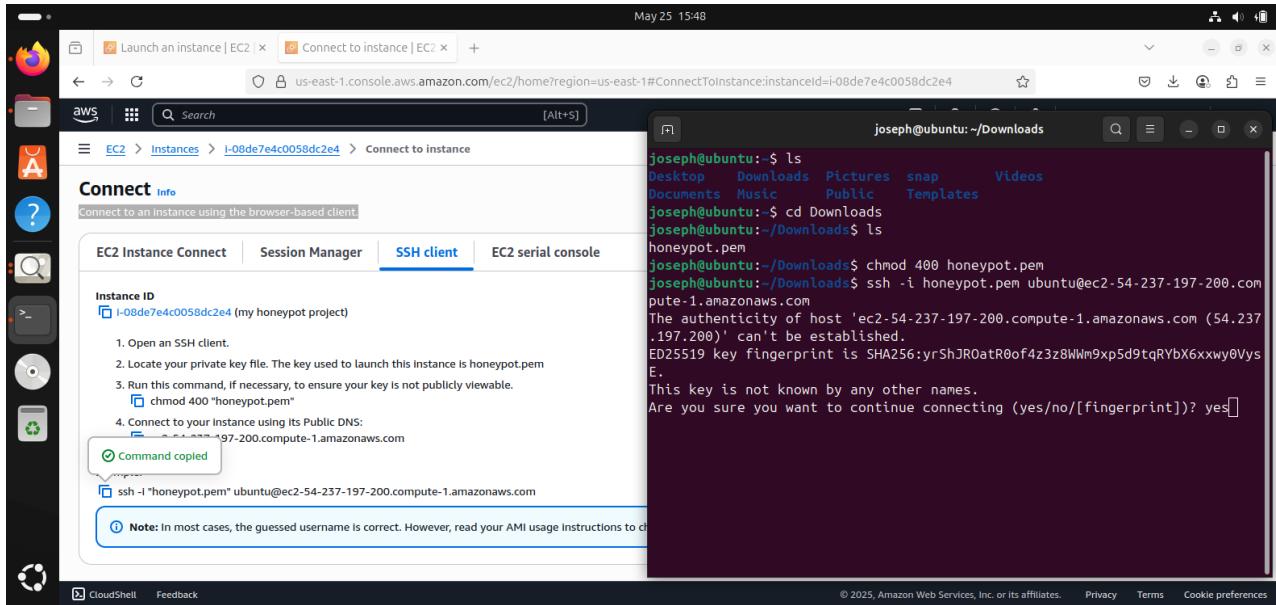
Click on the SSH client and open your terminal to follow the command instructions to connect the instance.



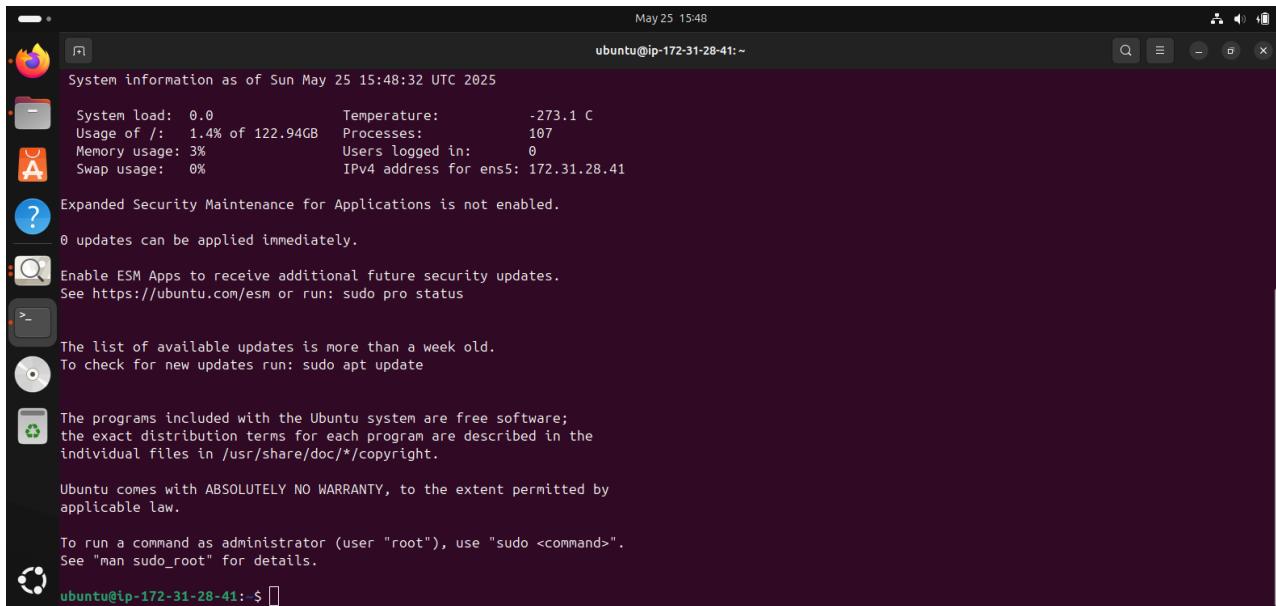
On your terminal change directory to downloads using ‘cd Downloads’, if you list the file inside downloads with ‘ls’ like I did above you will see the key pair you created while setting up the EC2 instance.



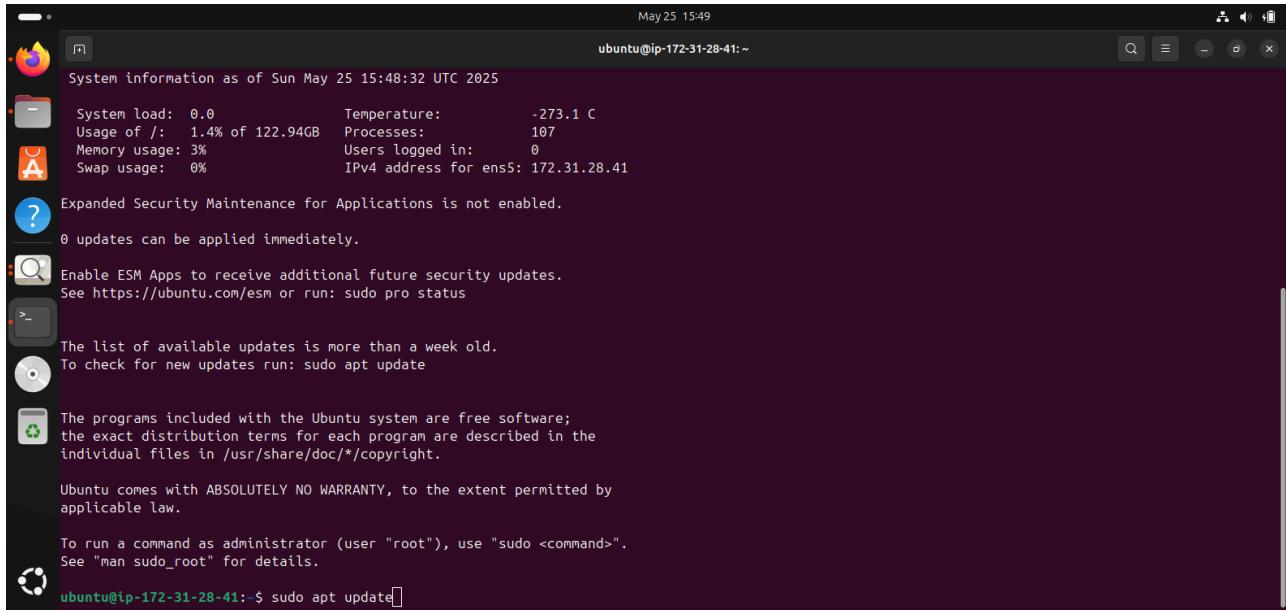
Following the instruction on the SSH client instance connection page put down the command chmod 400 and the key pair name you created, mine is ‘chmod 400 honeypot.pem’ then click enter.



After clicking enter you need to connect to your instance using the 4th command on the SSH instance connection page, copy the command on the page and past it on your terminal and click enter it will ask you to continue connection click 'y' to confirm (make sure you remove the quotation mark on the key pair of the command if it's not in the key pair saved on your file).



After clicking enter you will notice you will be login to the instance on your terminal just like mine in the picture above.



System information as of Sun May 25 15:48:32 UTC 2025

May 25 15:49
ubuntu@ip-172-31-28-41:~

System load: 0.0 Temperature: -273.1 C
Usage of /: 1.4% of 122.94GB Processes: 107
Memory usage: 3% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 172.31.28.41

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

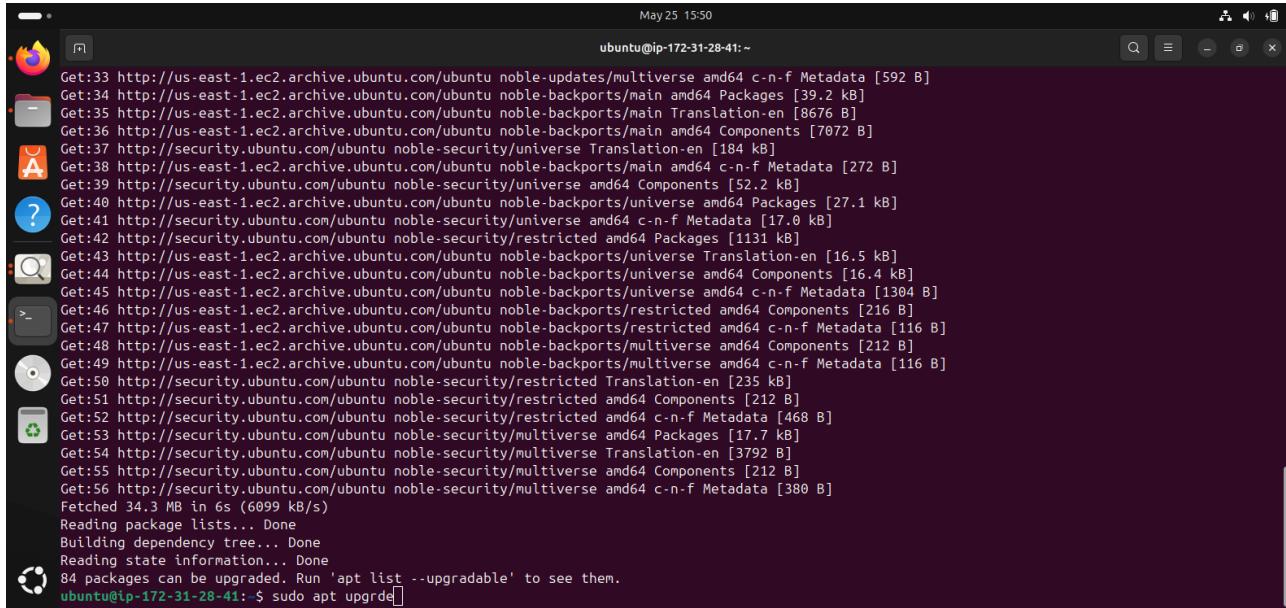
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-28-41: \$ sudo apt update

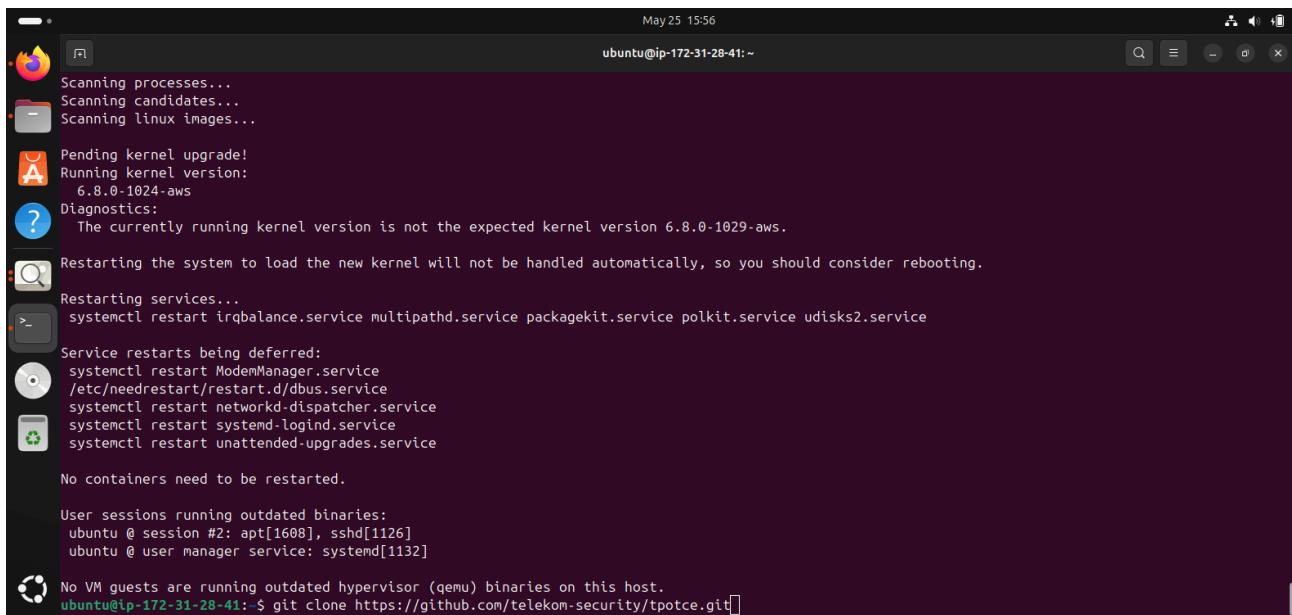
After logging in you need to update the system, to do that you need to run the command sudo apt update then click enter.



```
May 25 15:50  
ubuntu@ip-172-31-28-41:~
```

```
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [592 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [39.2 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main Translation-en [8676 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7072 B]
Get:37 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [184 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [272 B]
Get:39 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [27.1 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [17.0 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1131 kB]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [16.5 kB]
Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [16.4 kB]
Get:45 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1304 B]
Get:46 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:47 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:48 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:49 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [235 kB]
Get:51 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:52 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [468 B]
Get:53 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [17.7 kB]
Get:54 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [3792 B]
Get:55 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:56 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [380 B]
Fetched 34.3 MB in 6s (6099 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
84 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-28-41: $ sudo apt upgrade
```

After system has updated you need to upgrade the system, to do that write the command sudo apt upgrade then click enter, it will ask you to confirm update click 'y' to confirm.



Scanning processes...
Scanning candidates...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
6.8.0-1024-aws
Diagnostics:
The currently running kernel version is not the expected kernel version 6.8.0-1029-aws.

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

Restarting services...
systemctl restart irqbalance.service multipathd.service packagekit.service polkit.service udisks2.service

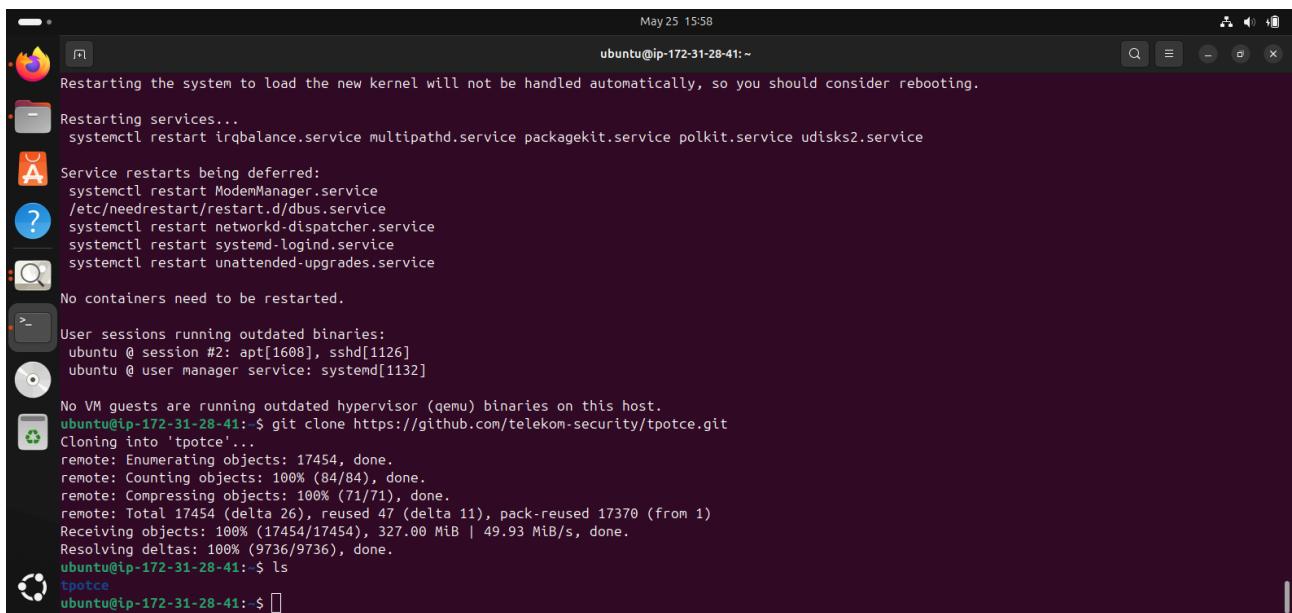
Service restarts being deferred:
systemctl restart ModemManager.service
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #2: apt[1608], sshd[1126]
ubuntu @ user manager service: systemd[1132]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-28-41: \$ git clone https://github.com/telekom-security/tpotce.git

After upgrading we have to clone the github t-pot repository and to do that type the command: <https://github.com/telekom-security/tpotce.git> then click enter.



Restaring the system to load the new kernel will not be handled automatically, so you should consider rebooting.

Restarting services...
systemctl restart irqbalance.service multipathd.service packagekit.service polkit.service udisks2.service

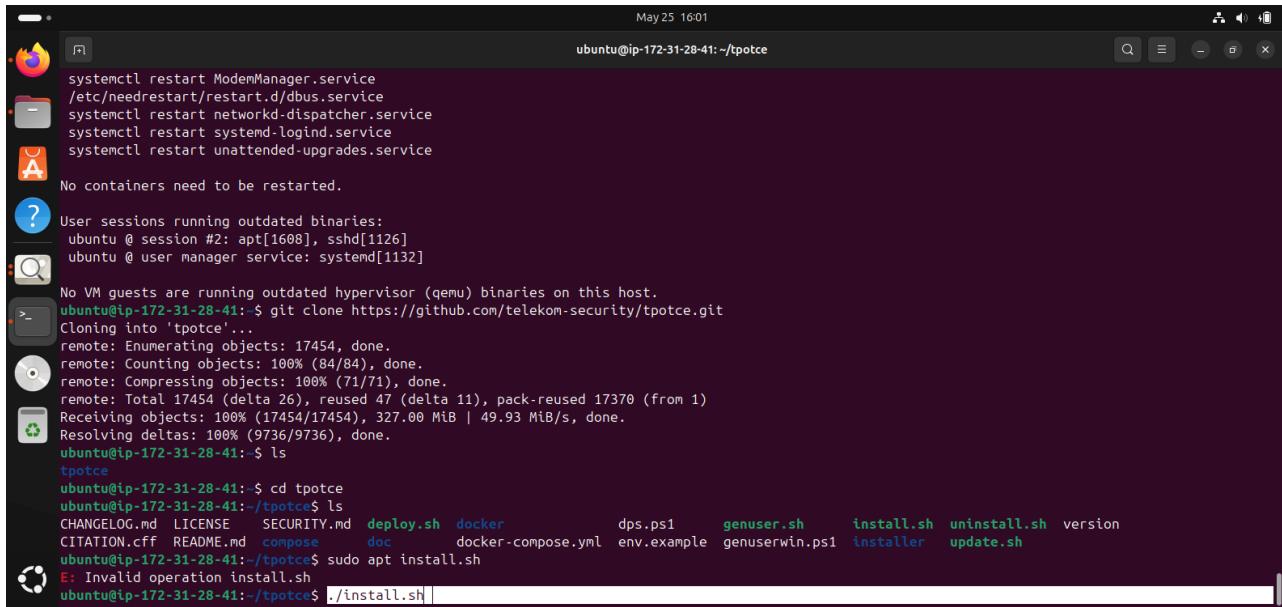
Service restarts being deferred:
systemctl restart ModemManager.service
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #2: apt[1608], sshd[1126]
ubuntu @ user manager service: systemd[1132]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-28-41: \$ git clone https://github.com/telekom-security/tpotce.git
Cloning into 'tpotce'...
remote: Enumerating objects: 17454, done.
remote: Counting objects: 100% (84/84), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 17454 (delta 26), reused 47 (delta 11), pack-reused 17370 (from 1)
Receiving objects: 100% (17454/17454), 327.00 MiB | 49.93 MiB/s, done.
Resolving deltas: 100% (9736/9736), done.
ubuntu@ip-172-31-28-41: \$ ls
tpotce
ubuntu@ip-172-31-28-41: \$

After cloning the github repository to list directories with the command 'ls' to see the newly cloned github repository.



```
May 25 16:01
ubuntu@ip-172-31-28-41:~/tpotce

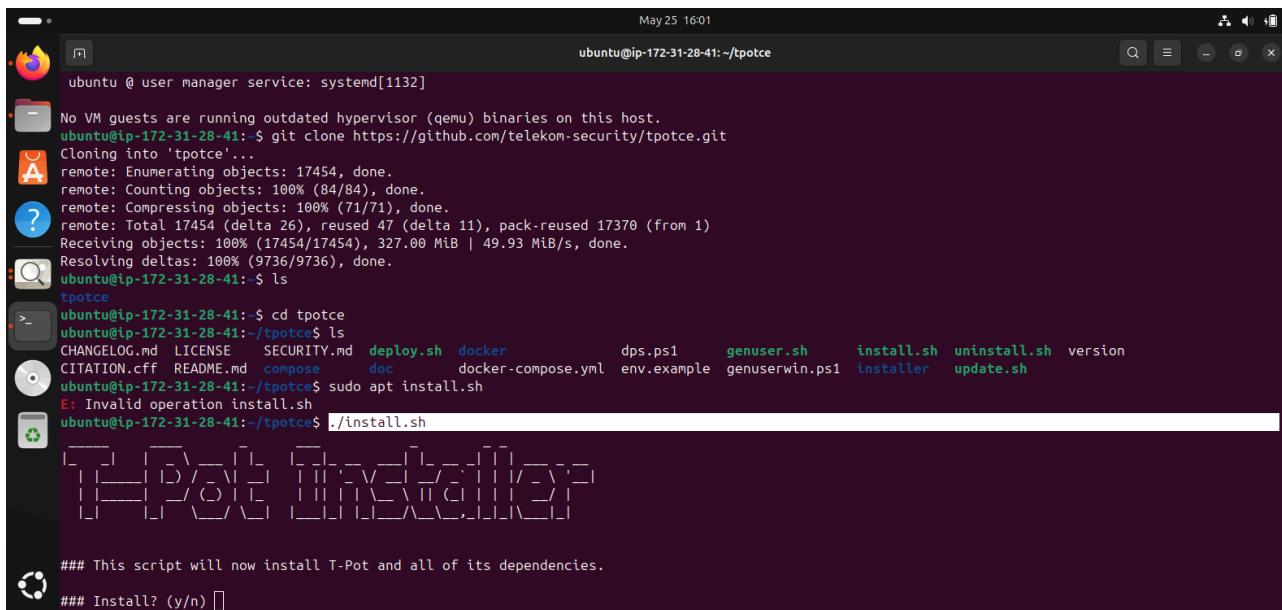
systemctl restart ModemManager.service
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #2: apt[1608], sshd[1126]
ubuntu @ user manager service: systemd[1132]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-28-41: $ git clone https://github.com/telekom-security/tpotce.git
Cloning into 'tpotce'...
remote: Enumerating objects: 17454, done.
remote: Counting objects: 100% (84/84), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 17454 (delta 26), reused 47 (delta 11), pack-reused 17370 (from 1)
Receiving objects: 100% (17454/17454), 327.00 MiB | 49.93 MiB/s, done.
Resolving deltas: 100% (9736/9736), done.
ubuntu@ip-172-31-28-41: $ ls
tpotce
ubuntu@ip-172-31-28-41: $ cd tpotce
ubuntu@ip-172-31-28-41:~/tpotce$ ls
CHANGELOG.md LICENSE SECURITY.md deploy.sh docker dps.ps1 genuser.sh install.sh uninstall.sh version
CITATION.cff README.md compose doc docker-compose.yml env.example genuserwin.ps1 installer update.sh
ubuntu@ip-172-31-28-41:~/tpotce$ sudo apt install.sh
E: Invalid operation install.sh
ubuntu@ip-172-31-28-41:~/tpotce$ ./install.sh
```

After listing the repository change directory to the tpotce file with the command ‘cd tpotce’ then click enter, there after list directories with the command ‘ls’ then install the tpot depository with the command ‘./install.sh’ then click enter.

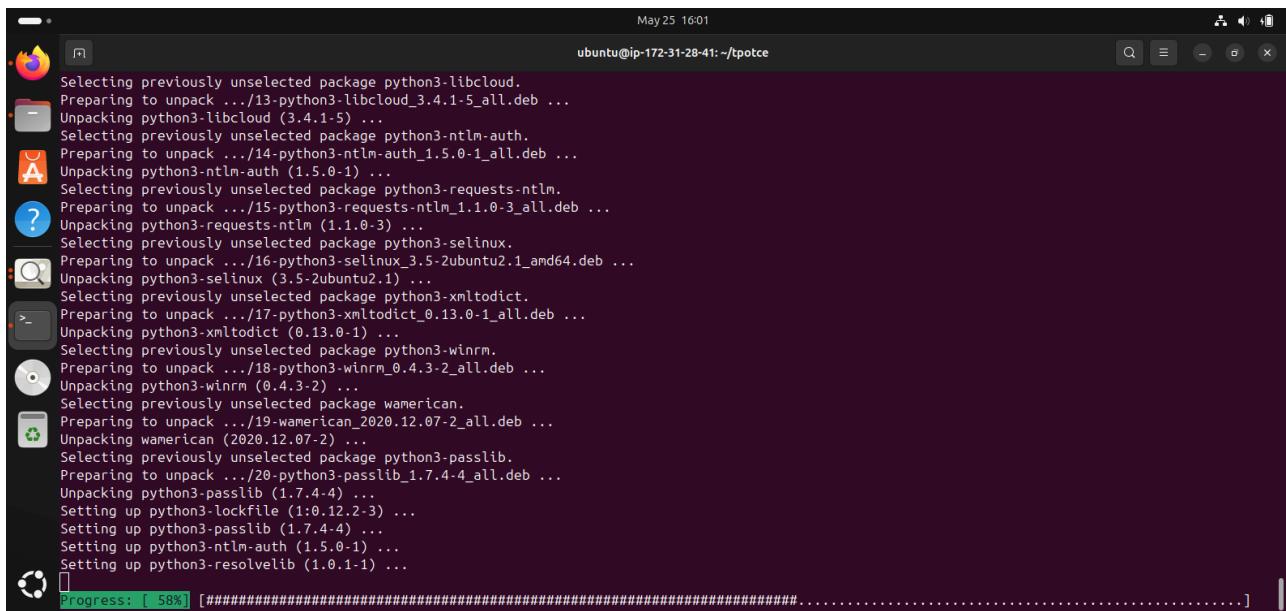


```
May 25 16:01
ubuntu@ip-172-31-28-41:~/tpotce

ubuntu @ user manager service: systemd[1132]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-28-41: $ git clone https://github.com/telekom-security/tpotce.git
Cloning into 'tpotce'...
remote: Enumerating objects: 17454, done.
remote: Counting objects: 100% (84/84), done.
remote: Compressing objects: 100% (71/71), done.
remote: Total 17454 (delta 26), reused 47 (delta 11), pack-reused 17370 (from 1)
Receiving objects: 100% (17454/17454), 327.00 MiB | 49.93 MiB/s, done.
Resolving deltas: 100% (9736/9736), done.
ubuntu@ip-172-31-28-41: $ ls
tpotce
ubuntu@ip-172-31-28-41: $ cd tpotce
ubuntu@ip-172-31-28-41:~/tpotce$ ls
CHANGELOG.md LICENSE SECURITY.md deploy.sh docker dps.ps1 genuser.sh install.sh uninstall.sh version
CITATION.cff README.md compose doc docker-compose.yml env.example genuserwin.ps1 installer update.sh
ubuntu@ip-172-31-28-41:~/tpotce$ sudo apt install.sh
E: Invalid operation install.sh
ubuntu@ip-172-31-28-41:~/tpotce$ ./install.sh
[Progress Bar: 0% to 100%]
### This script will now install T-Pot and all of its dependencies.
### Install? (y/n) [
```

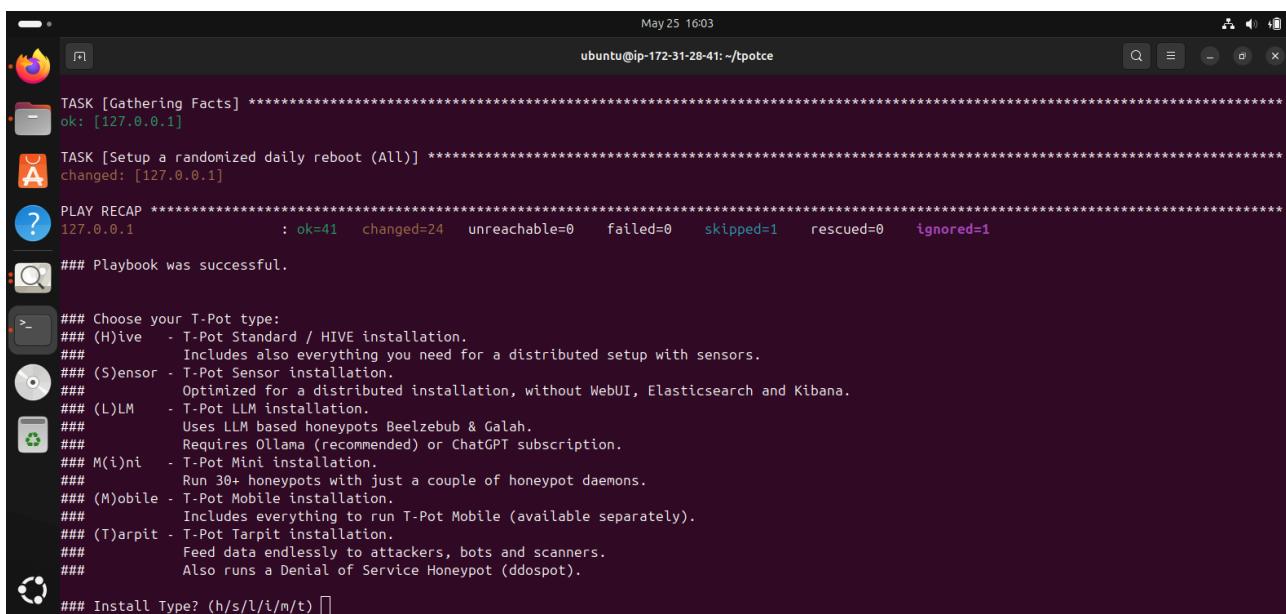
You will be asked if you want to install you should type ‘y’ to continue installation.



ubuntu@ip-172-31-28-41:~/tpotce

```
Selecting previously unselected package python3-libcloud.
Preparing to unpack .../13-python3-libcloud_3.4.1-5_all.deb ...
Unpacking python3-libcloud (3.4.1-5) ...
Selecting previously unselected package python3-ntlm-auth.
Preparing to unpack .../14-python3-ntlm-auth_1.5.0-1_all.deb ...
Unpacking python3-ntlm-auth (1.5.0-1) ...
Selecting previously unselected package python3-requests.
Preparing to unpack .../15-python3-requests_ntlm_1.1.0-3_all.deb ...
Unpacking python3-requests_ntlm (1.1.0-3) ...
Selecting previously unselected package python3-selinux.
Preparing to unpack .../16-python3-selinux_3.5-2ubuntu2.1_amd64.deb ...
Unpacking python3-selinux (3.5-2ubuntu2.1) ...
Selecting previously unselected package python3-xmldict.
Preparing to unpack .../17-python3-xmldict_0.13.0-1_all.deb ...
Unpacking python3-xmldict (0.13.0-1) ...
Selecting previously unselected package python3-winrm.
Preparing to unpack .../18-python3-winrm_0.4.3-2_all.deb ...
Unpacking python3-winrm (0.4.3-2) ...
Selecting previously unselected package wamerican.
Preparing to unpack .../19-wamerican_2020.12.07-2_all.deb ...
Unpacking wamerican (2020.12.07-2) ...
Selecting previously unselected package python3-passlib.
Preparing to unpack .../20-python3-passlib_1.7.4-4_all.deb ...
Unpacking python3-passlib (1.7.4-4) ...
Setting up python3-lockfile (1:0.12.2-3) ...
Setting up python3-passlib (1.7.4-4) ...
Setting up python3-ntlm-auth (1.5.0-1) ...
Setting up python3-resolvelib (1.0.1-1) ...
Progress: [ 58%] [#####.....]
```

Installation process...



```
ubuntu@ip-172-31-28-41:~/tpotce
```

```
TASK [Gathering Facts] *****
ok: [127.0.0.1]

TASK [Setup a randomized daily reboot (All)] *****
changed: [127.0.0.1]

PLAY RECAP *****
127.0.0.1 : ok=41   changed=24   unreachable=0   failed=0   skipped=1   rescued=0   ignored=1

### Playbook was successful.

-> #### Choose your T-Pot type:
    ### (H)ive - T-Pot Standard / HIVE installation.
    ### Includes also everything you need for a distributed setup with sensors.
    ### (S)ensor - T-Pot Sensor installation.
    ### Optimized for a distributed installation, without WebUI, Elasticsearch and Kibana.
    ### (L)LM - T-Pot LLM installation.
    ### Uses LLM based honeypots Beelzebub & Galah.
    ### Requires Ollama (recommended) or ChatGPT subscription.
    ### M(i)ni - T-Pot Mini installation.
    ### Run 30+ honeypots with just a couple of honeypot daemons.
    ### (M)oile - T-Pot Mobile installation.
    ### Includes everything to run T-Pot Mobile (available separately).
    ### (T)arpit - T-Pot Tarpit installation.
    ### Feed data endlessly to attackers, bots and scanners.
    ### Also runs a Denial of Service Honeypot (ddospot).

### Install Type? (h/s/l/i/m/t) 
```

Before installation complete you will be ask to choose honeypot type, you have to choose –T-pot standard / hive installation, to do that simply type in ‘h’ and click enter.

```

May 25 16:07
ubuntu@ip-172-31-28-41: ~/tpotce

✓ sentrypeer Pulled
✓ heralding Pulled
✓ elasticsearch Pulled
✓ suricata Pulled
✓ map_web Pulled
✓ fatt Pulled
✓ h0neytr4p Pulled
✓ elasticpot Pulled
✓ redishoneypot Pulled
✓ logstash Pulled
✓ ewsposter Pulled
✓ pof Pulled
✓ ciscoasa Pulled

>_ ## Please review for possible honeypot port conflicts.
## While SSH is taken care of, other services such as
## SMTP, HTTP, etc. might prevent T-Pot from starting.

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State      User   Inode PID/Program name
tcp        0      0 0.0.0.0:64295            0.0.0.0:*          LISTEN     0      51388 23648/sshd: /usr/sb
tcp6       0      0 :::64295                :::*                  LISTEN     0      51390 23648/sshd: /usr/sb
udp        0      0 172.31.28.41:68          0.0.0.0:*          998       17378 5396/systemd-networ
udp        0      0 127.0.0.1:323            0.0.0.0:*          0        7598   764/chrony
udp6       0      0 ::1:323                 :::*                  0        7599   764/chrony

## Done. Please reboot and re-connect via SSH on tcp/64295.

ubuntu@ip-172-31-28-41:~/tpotce$ 
```

After choosing honeypot type installation will complete and you will be asked to reboot and reconnect, to do that type in the command ‘sudo reboot’ and click enter.

```

May 25 16:08
joseph@ubuntu: ~/Downloads

✓ elasticpot Pulled
✓ redishoneypot Pulled
✓ logstash Pulled
✓ ewsposter Pulled
✓ pof Pulled
✓ ciscoasa Pulled

? ## Please review for possible honeypot port conflicts.
## While SSH is taken care of, other services such as
## SMTP, HTTP, etc. might prevent T-Pot from starting.

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State      User   Inode PID/Program name
tcp        0      0 0.0.0.0:64295            0.0.0.0:*          LISTEN     0      51388 23648/sshd: /usr/sb
tcp6       0      0 :::64295                :::*                  LISTEN     0      51390 23648/sshd: /usr/sb
udp        0      0 172.31.28.41:68          0.0.0.0:*          998       17378 5396/systemd-networ
udp        0      0 127.0.0.1:323            0.0.0.0:*          0        7598   764/chrony
udp6       0      0 ::1:323                 :::*                  0        7599   764/chrony

## Done. Please reboot and re-connect via SSH on tcp/64295.

ubuntu@ip-172-31-28-41:~/tpotce$ sudo reboot
Broadcast message from root@ip-172-31-28-41 on pts/1 (Sun 2025-05-25 16:08:00 UTC):
The system will reboot now!
joseph@ubuntu:~/Downloads$ 
```

After reboot you will be kicked out as you can see in the picture above, now we need to setup some security protocol on our AWS EC2 instance.

The screenshot shows the AWS EC2 Instances dashboard. At the top, there's a search bar and navigation links for 'Launch an instance | EC2' and 'Instances | EC2 | us-east-1'. The main table lists one instance: 'my honeypot' (ID: i-08de7e4c0058dc2e4), which is 'Running' and has an 't3.large' instance type. It has a status of '3/3 checks passed' and is in the 'us-east-1c' availability zone. The public IPv4 DNS is 'ec2-54-237-197-200.co...' and the IP address is '54.237.197.200'. Below the table, a specific instance detail page is shown for 'i-08de7e4c0058dc2e4 (my honeypot project)'. The 'Security' tab is selected, showing 'Security details' (IAM Role: 'Owner ID 259959202252') and 'Inbound rules' (one rule for port 22). The 'Launch time' is listed as 'Sun May 25 2025 15:40:05 GMT+0000 (Coordinated Universal Time)'.

On the EC2 instance dashboard click on security to start setting security protocol setting.

The screenshot shows the 'Instance details' page for the instance 'i-08de7e4c0058dc2e4'. The 'Security' tab is selected. Under 'Security details', it shows the IAM Role 'Owner ID 259959202252'. Under 'Inbound rules', there is one rule: 'sgr-02fee540e93c0de9d' allowing port 22 from 0.0.0.0/0 to the security group 'launch-wizard-1'. Under 'Outbound rules', there is one rule: 'sgr-0032d83093ce4919a' allowing all ports to 0.0.0.0/0 from the security group 'launch-wizard-1'.

After clicking on security click on the blue spot which Is saying ‘lunch-wizard’

The screenshot shows the AWS EC2 Security Groups page. On the left sidebar, under the 'Instances' section, there is a 'Launch Wizard' icon. The main content area displays a table titled 'Security Groups (1)'. The table has columns for Name, Security group ID, Security group name, VPC ID, and Description. One row is listed: Name is 'sg-0d2f400ec3a3b8715', Security group ID is 'vpc-013f34b7b809248d0', Security group name is 'launch-wizard-1', and Description is 'launch-wizard-1'. Below the table, a section titled 'Select a security group' is visible.

After that step click on the blue spot that starts with ‘vpc-’ to get to the page to set inbound rules.

The screenshot shows the 'Edit inbound rules' page for the security group 'sg-0d2f400ec3a3b8715 - launch-wizard-1'. The page title is 'ModifyInboundSecurityGroupRules:securityGroupId=sg-0d2f400ec3a3b8715'. The main content area is titled 'Inbound rules' and contains a table with three rows. The columns are: Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. The 'Type' column shows 'Custom TCP' for all three rows. The 'Protocol' column shows 'TCP' for all three rows. The 'Port range' column shows '0' for all three rows. The 'Source' column shows 'Custom' for all three rows. The 'Description - optional' column is empty for all three rows. At the bottom of the table, there are three 'Delete' buttons. Below the table, there are buttons for 'Add rule', 'Cancel', 'Preview changes', and 'Save rules'.

You are going to setup three security protocol which will be displayed and explain below.

The screenshot shows the AWS CloudShell interface with the AWS logo. The title bar reads "May 25 16:14". The main content area is titled "Edit inbound rules" and displays three inbound security group rules:

- Rule 1: Type: Custom TCP, Protocol: TCP, Port range: 64295, Source: My IP, Description: 197.210.28.60/32
- Rule 2: Type: Custom TCP, Protocol: TCP, Port range: 64297, Source: My IP, Description: 197.210.28.60/32
- Rule 3: Type: Custom TCP, Protocol: TCP, Port range: 1 - 64000, Source: Anywhere (IPv4), Description: 0.0.0.0/0

A warning message at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your Instance. We recommend setting security group rules to allow access from known IP addresses only." There is also a note about CloudShell and feedback.

On the type leave the three rules as Custom TCP, on the port range the first port should be set as 64295, this is the SSH port that will be used to login the command base interface of the honeypot, on the second port range input 64297, this port will be used to access the graphical interface, and the third port should be set from 1-64000, on the source, the first source should be set as ‘My IP’ the second source should also be set as ‘My IP’, then the last source should be set for ‘Anywhere ipv4’, then click on save rules.

```

joseph@ubuntu:~/Downloads
-----
✓ elasticpot Pulled
✓ redishoneypot Pulled
✓ logstash Pulled
✓ ewsposter Pulled
✓ p0f Pulled
✓ ciscoasa Pulled

### Please review for possible honeypot port conflicts.
### While SSH is taken care of, other services such as
### SMTP, HTTP, etc. might prevent T-Pot from starting.

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      User       Inode      PID/Program name
tcp        0      0 0.0.0.0:64295           0.0.0.0:*          LISTEN     0          51388     23648/sshd: /usr/sbin/sshd
tcp6       0      0 :::64295              :::*                  LISTEN     0          51390     23648/sshd: /usr/sbin/sshd
udp        0      0 172.31.28.41:68         0.0.0.0:*          998        17378     5396/systemd-network
udp        0      0 127.0.0.1:323          0.0.0.0:*          0          7598      764/chrony
udp6       0      0 ::1:323              :::*                  0          7599      764/chrony

### Done. Please reboot and re-connect via SSH on tcp/64295.

ubuntu@ip-172-31-28-41:~/tptce$ sudo reboot
Broadcast message from root@ip-172-31-28-41 on pts/1 (Sun May 25 16:08:00 UTC):
The system will reboot now!
ubuntu@ip-172-31-28-41:~/tptce$ client_loop: send disconnect: Broken pipe
joseph@ubuntu:~/Downloads$ ssh -i honeypot.pem ubuntu@ec2-54-237-197-200.compute-1.amazonaws.com -p 64295

```

After that go back to your terminal and login with the SSH instance connection command but at the end you will add -p 64295 which is your command base interface login port.

```

May 25 16:16
ubuntu@ip-172-31-28-41:~$ client_loop: send disconnect: Broken pipe
joseph@ubuntu:~/Downloads$ ssh -i honeypot.pem ubuntu@ec2-54-237-197-200.compute-1.amazonaws.com -p 64295
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sun May 25 16:16:43 UTC 2025

System load: 0.4 Temperature: -273.1 C
Usage of /: 8.5% of 122.94GB Processes: 317
Memory usage: 87% Users logged in: 0
Swap usage: 0% IPv4 address for ens5: 172.31.28.41

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun May 25 15:48:33 2025 from 197.210.28.60
ubuntu@ip-172-31-28-41:~$ 

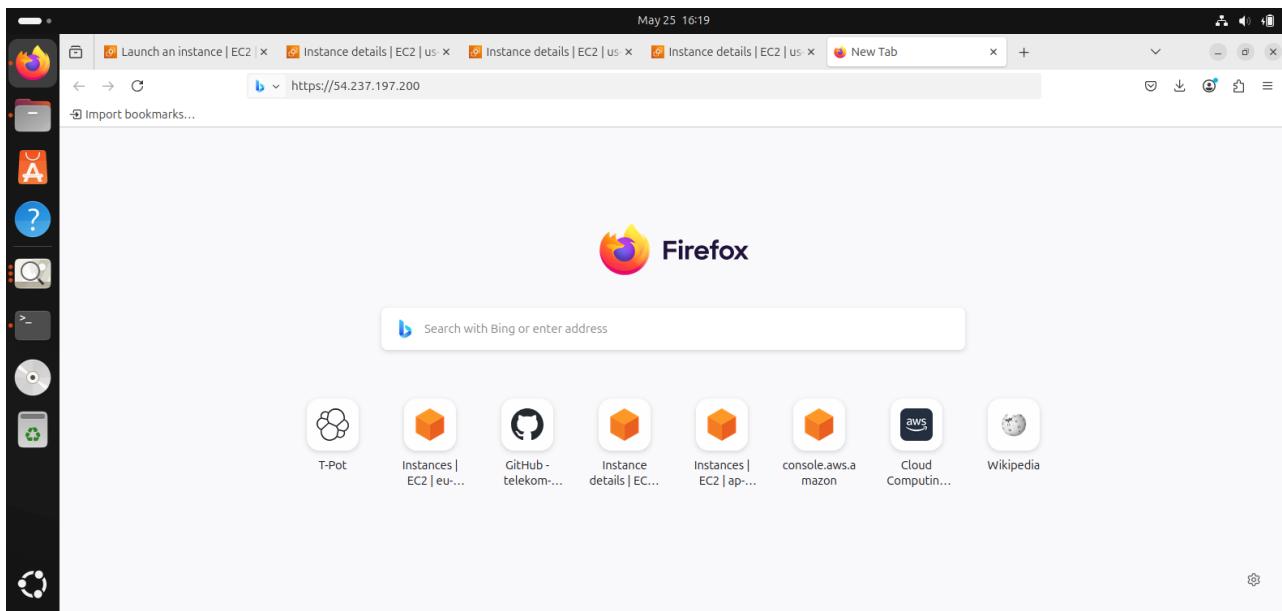
```

As you can see in the picture attach above the honeypot has been successfully login in the command base interface, now we have to go to our browser and login the graphical interface.

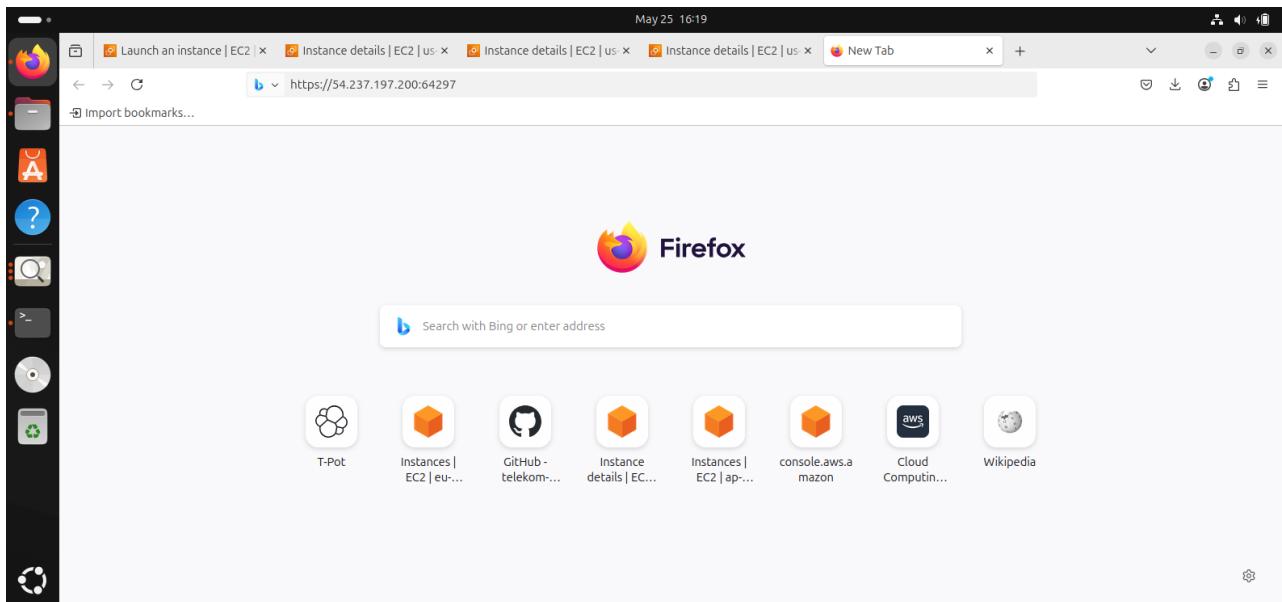
The screenshot shows the AWS EC2 Instances page for the instance `i-08de7e4c0058dc2e4`. The left sidebar shows navigation options like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. The main content area displays the instance summary for the honeypot project. Key details include:

- Instance ID:** `i-08de7e4c0058dc2e4`
- Public IPv4 address:** `54.237.197.200`
- Instance state:** `Running`
- Private IP DNS name (IPv4 only):** `ip-172-31-28-41.ec2.internal`
- Instance type:** `t3.large`
- VPC ID:** `vpc-013f34b7b809248d0`
- Subnet ID:** `subnet-0e718c826a146bbc0`

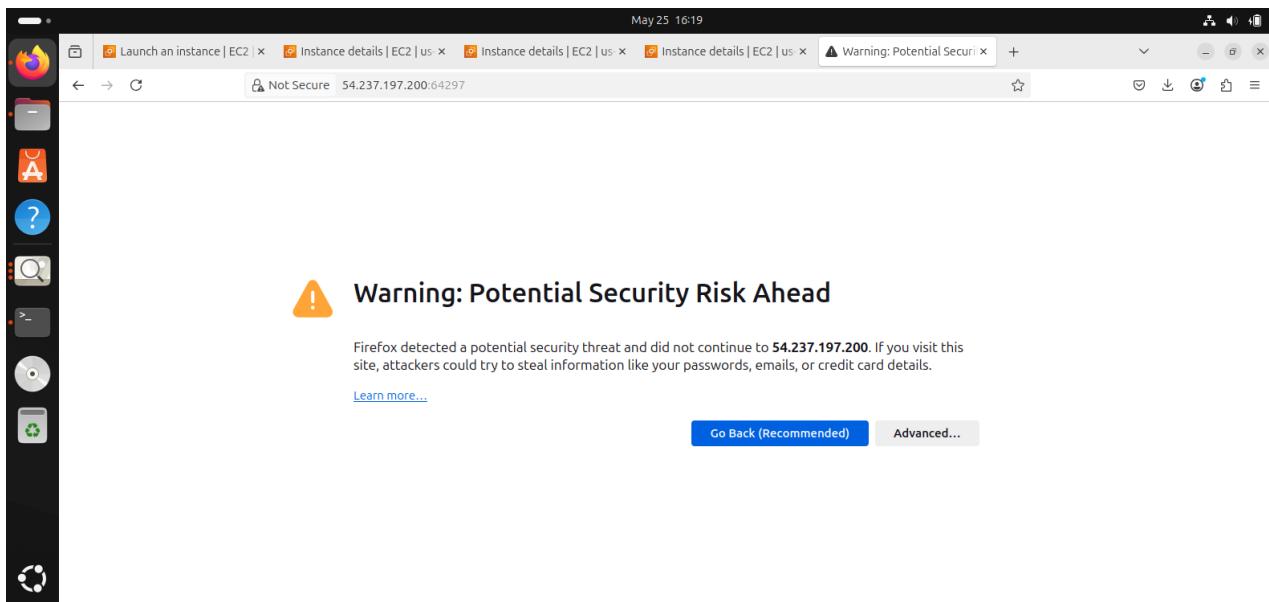
To be able to login our honeypot on the graphical interface we need to copy our instance public IP address and it can be found inside the EC2 instance as seen in the picture above.



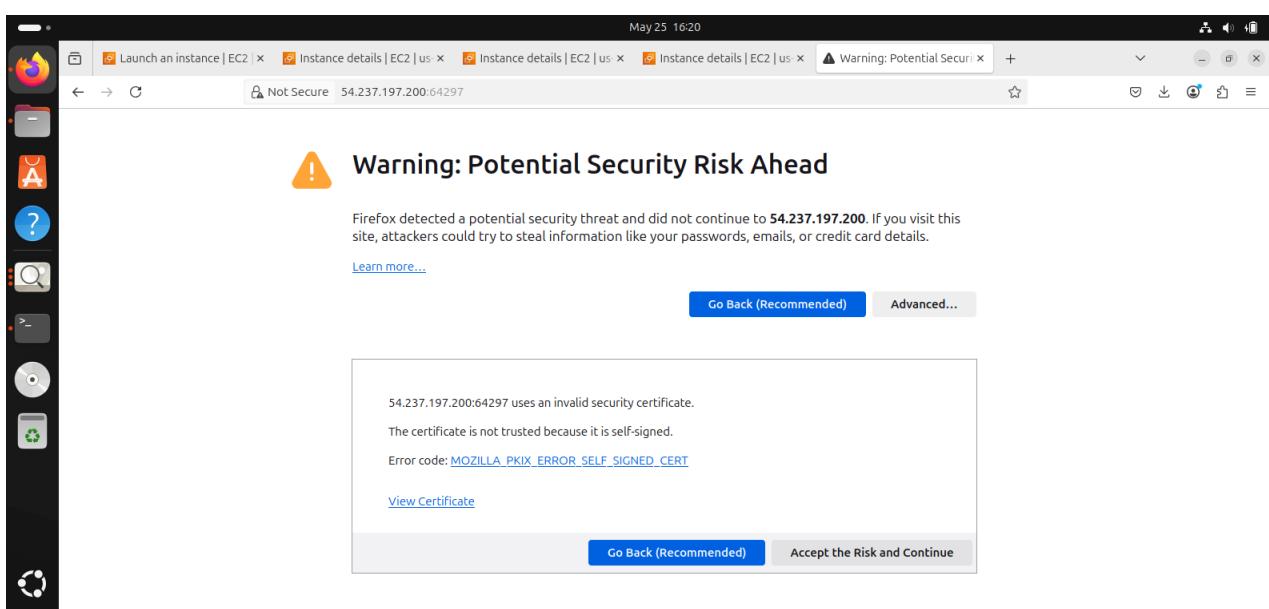
After coping the public address type: `https://` then past the pubic address you copied just as its seen above.



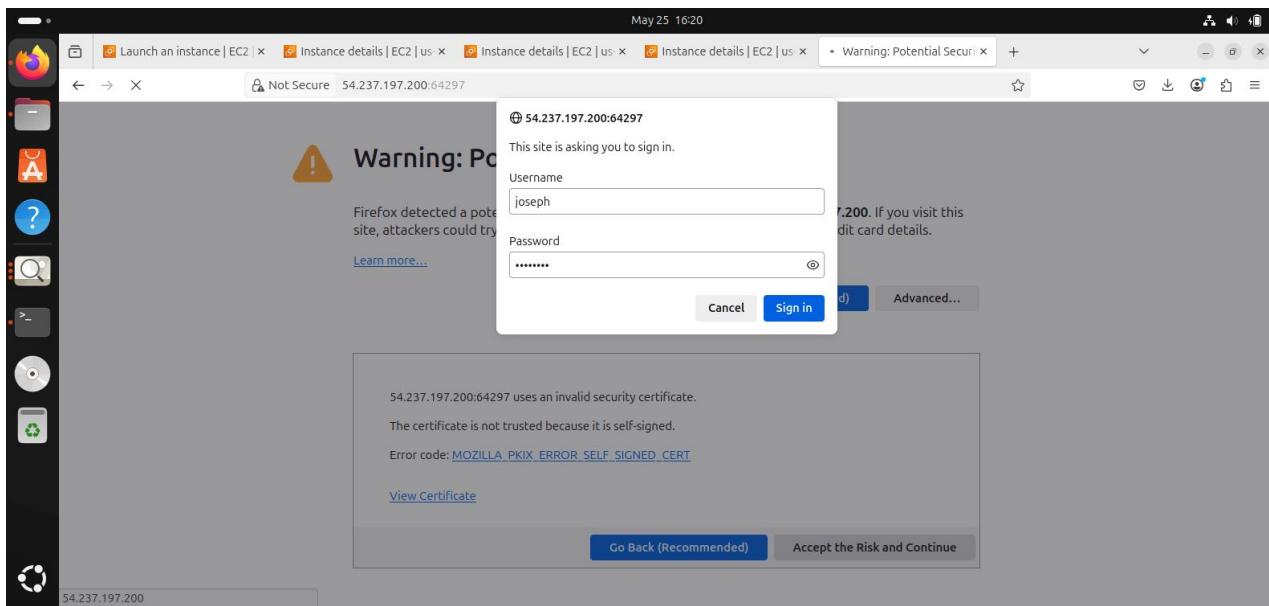
After that add : and the graphical port protocol 64297 then click enter.



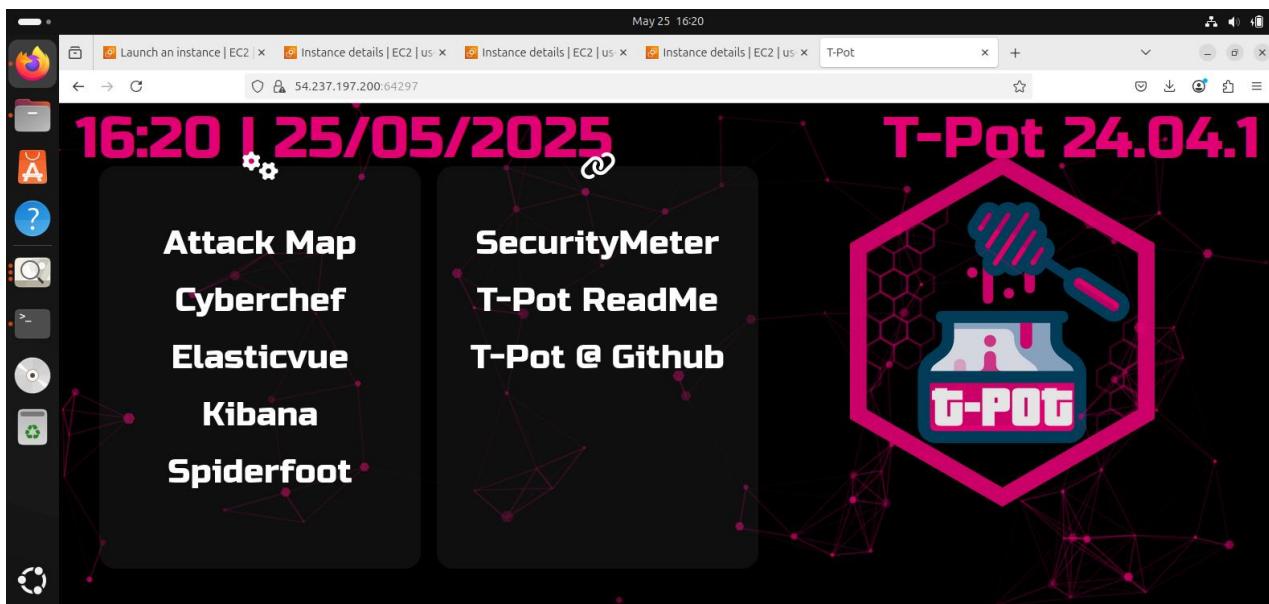
You will get a security risk alert, click on advance...



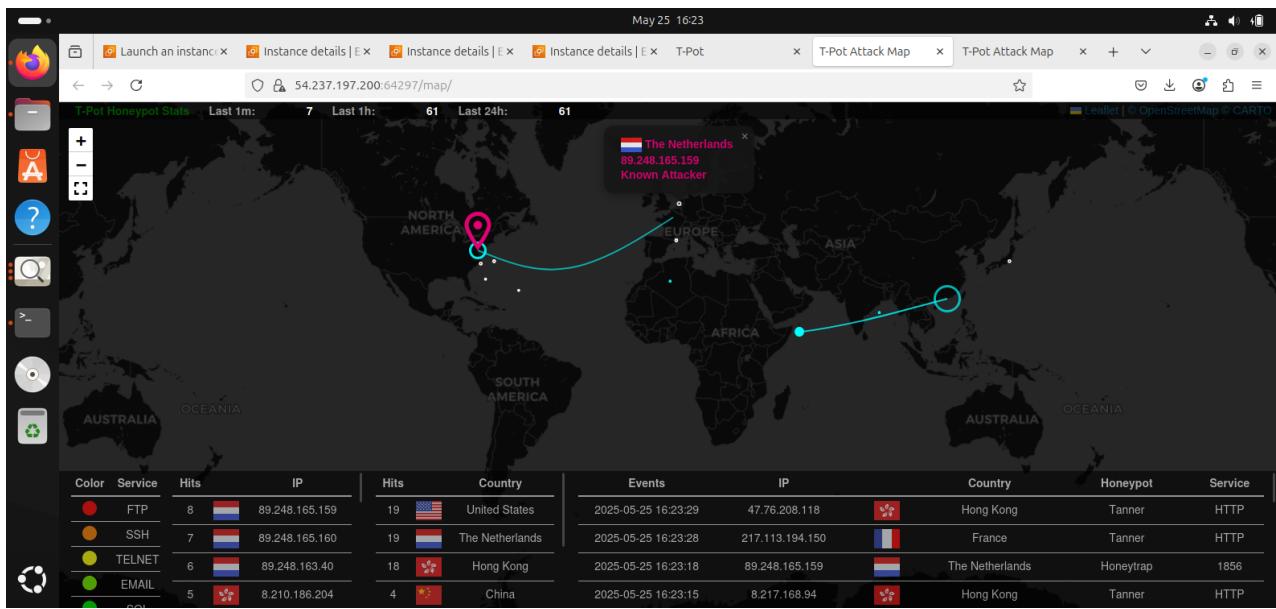
Accept the risk and continue.



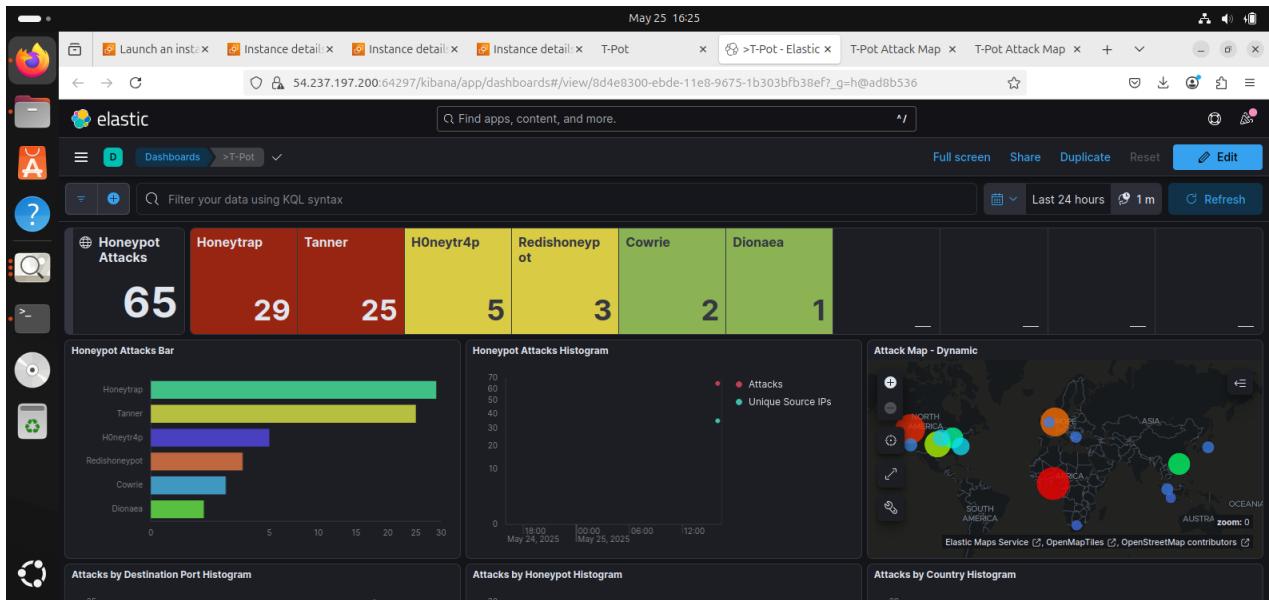
While installing the honeypot you were asked to create a password, sign in with that password.



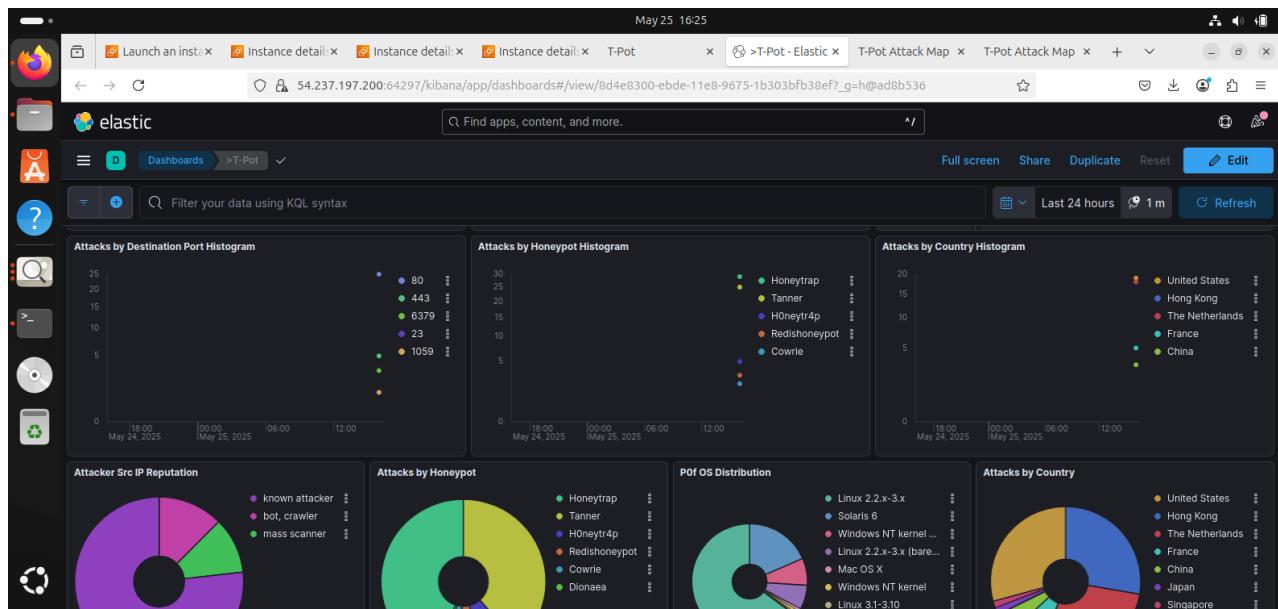
This is the graphical interface of our honeypot, the T-pot honeypot is a type of honeypot that consists of different honeypots and other features such as cryptographic features.



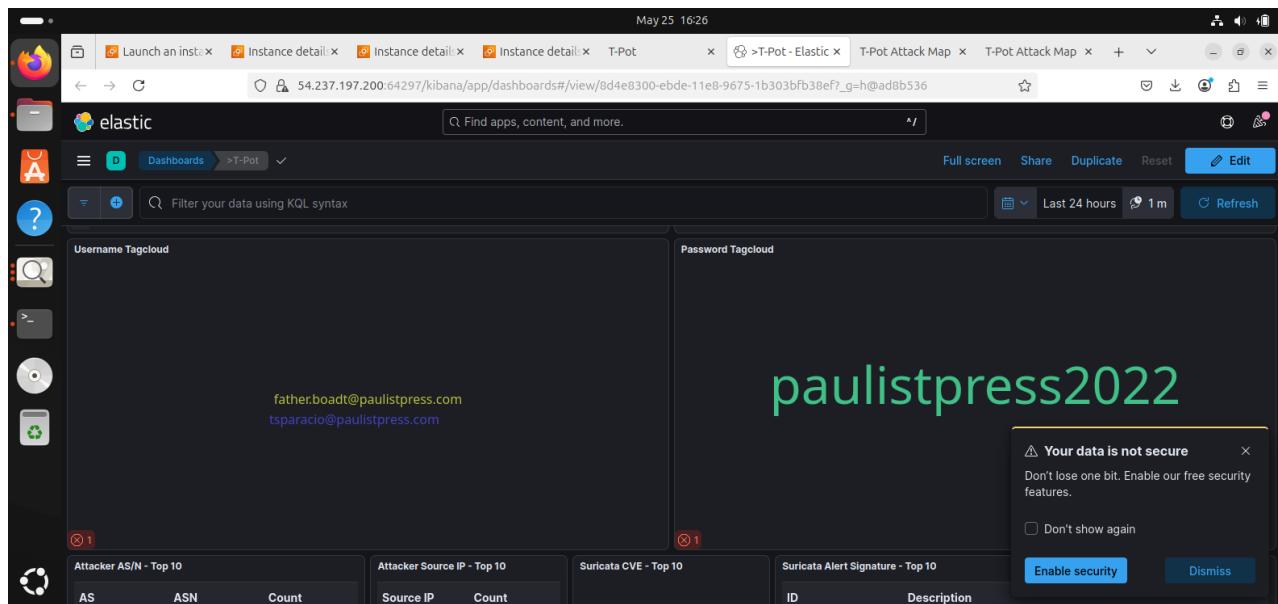
This is the attackers map, this allows us to monitor how attackers get into systems, the blue line on the map signify attacker trying to get into our system, we can also get their IP address and country they are attacking from on the attacker map.



This is kibana it is a dashboard that analyzes and record the attack on the different honey pot on the t-pot platform, scrolling down we can also see all the attackers IP addresses and also the ports they are logging in from, as well as getting the password and username they tried using to get into the honeypot.



This is a more break down of the attacked been carried out on our honeypot, this include attacker distribution, and their country.



This is the attempted email and password the hacker try using to get in.

This screenshot shows a Kibana dashboard titled 'T-Pot' with several panels:

- Attacker AS/N - Top 10:**

AS	ASN	Count
45102	Alibaba US Techr	26
202425	IP Volume inc	18
210743	Babbar SAS	12
4134	Chinanet	5
6939	HURRICANE	5
398324	CENSYS-ARIN-0	4
8075	MICROSOFT-COI	3
396982	GOOGLE-CLOUD	3
14618	AMAZON-AES	2
- Attacker Source IP - Top 10:**

Source IP	Count
89.248.165.15	7
89.248.165.16	6
89.248.163.40	5
8.210.185.180	4
8.210.186.204	4
125.67.236.4	3
217.113.194.14	3
217.113.194.15	3
8.221.141.186	3
- Suricata CVE - Top 10:** No results found.
- Suricata Alert Signature - Top 10:**

ID	Description
2101852	GPL WEB_SERVER robots.txt access
2010517	ET WEB_SERVER Possible HTTP 404 XSS Attempt (Local Source)
2402000	ET DROP Dshield Block Listed Source group 1
2210061	SURICATA STREAM ⚠ Your data is not secure
2009582	ET SCAN NMAP -sC
2002752	ET INFO Reserved I
2210041	SURICATA STREAM
2260002	SURICATA Aplayer
2400051	ET DROP Spamhaus DRO Listed Traffic inbound group 32

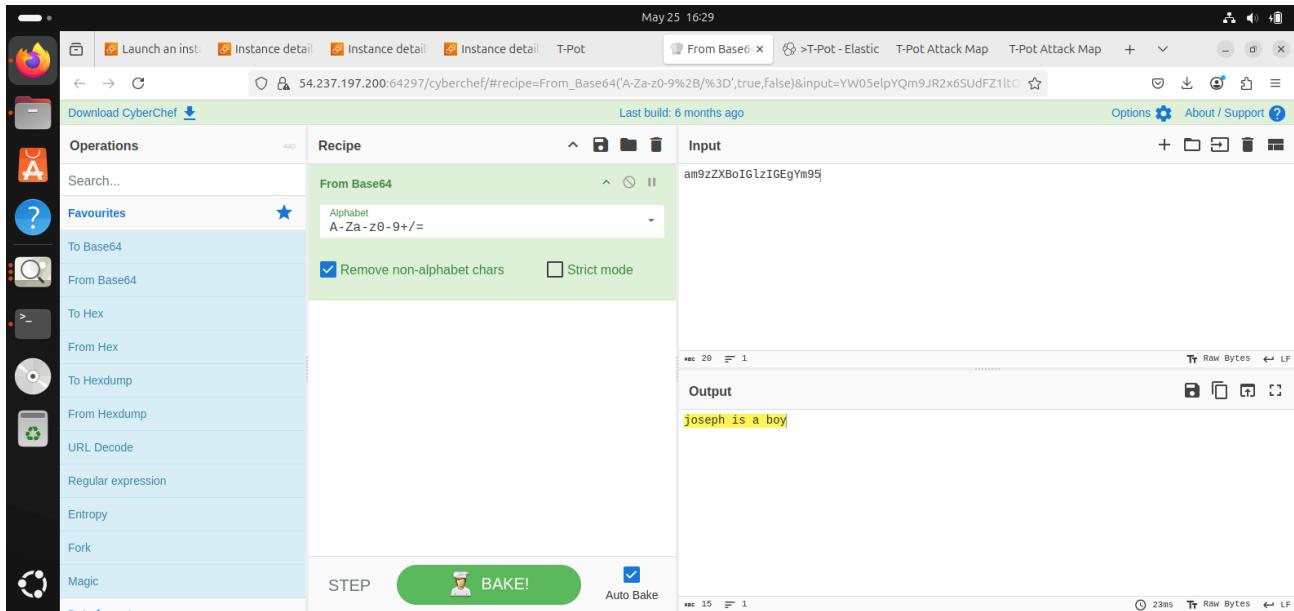
This is the details arrangement of the attacker details which include their IP address their ID and their AS and ASN.

This screenshot shows the CyberChef interface with the following configuration:

- Operations:** To Base64
- Input:** joseph is a boy
- Output:** am9zZXBoIGlzIGEgYm95

The interface includes a sidebar with various encoding and decoding options like To Hex, From Hex, To Base64, From Base64, etc.

This is the in-built spider-foot on this t-pot honeypot, it's a feature that can be used to encrypt and decrypt words, above is a sentence that has just been encrypted.



This above picture shows the decryption of same statement that was encrypted.

This document has presented a structured and detailed methodology for the design, implementation, and deployment of a honeypot system. Through a systematic, step-by-step approach, it has outlined the necessary technical configurations, tools, and best practices required to create an effective honeypot capable of capturing malicious activity and providing insights into attacker methodologies.