**Aim:**

To implement and configure Greenbone Vulnerability Management (GVM) for effective monitoring and scanning of network vulnerabilities, ensuring proactive detection of security issues and improving overall cybersecurity posture.

**Objectives:**

* To install and configure Greenbone GVM for vulnerability scanning of hosts and network infrastructure.
* To enable real-time alerting for detected vulnerabilities and critical security risks.
* To monitor systems for weaknesses such as outdated software, misconfigurations, and open ports.
* To integrate additional feeds and plugins for comprehensive vulnerability coverage.
* To enhance IT security reliability through proactive vulnerability management and threat detection.

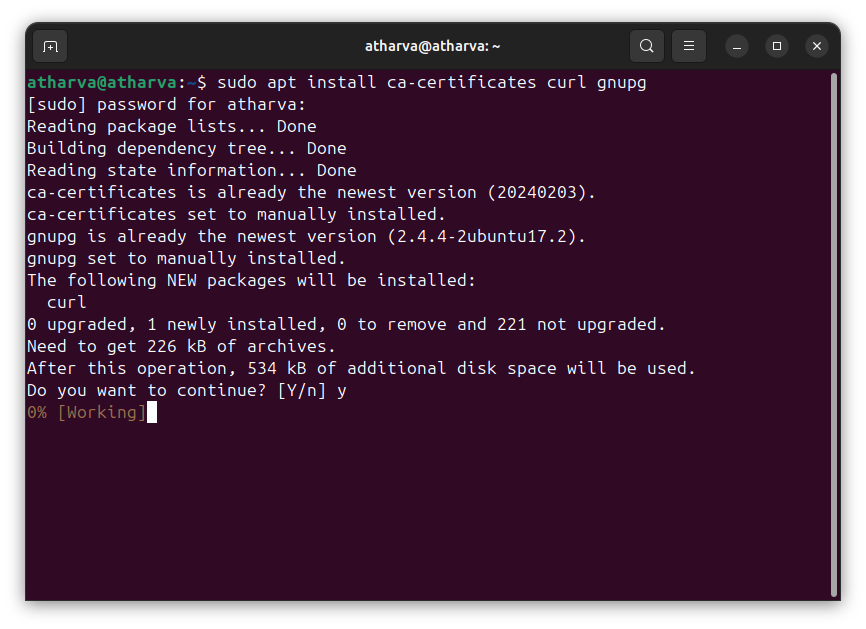
**Tools Used:**

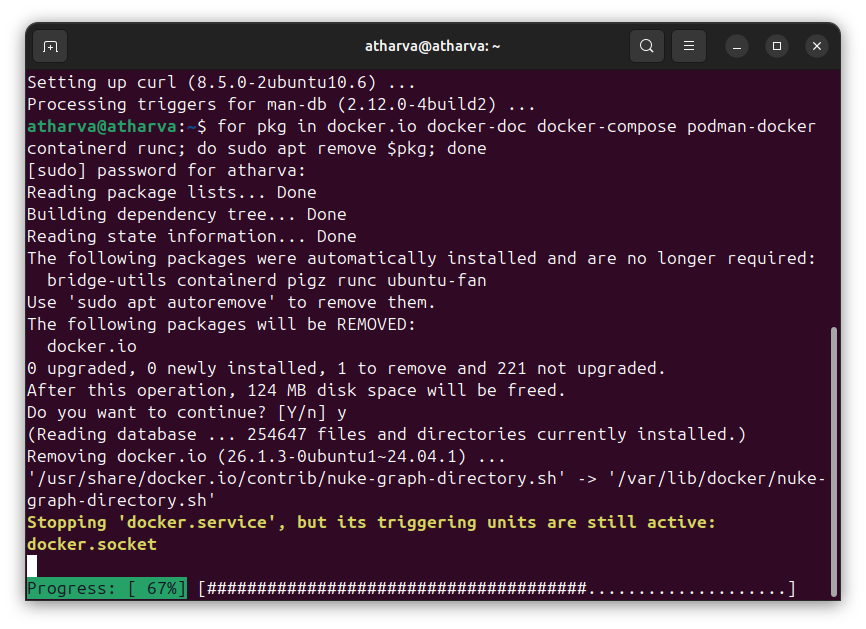
**VirtualBox**

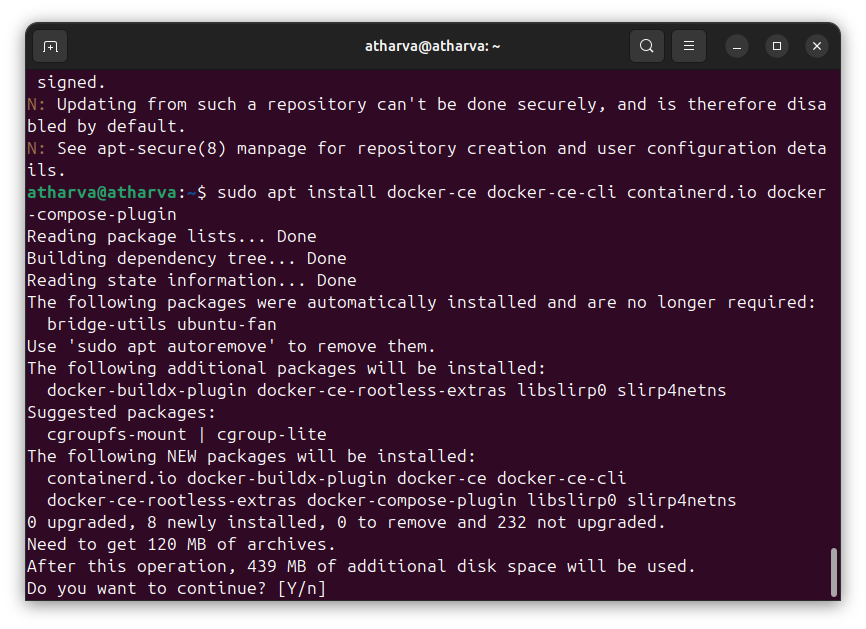
**Ubuntu (Linux OS)**

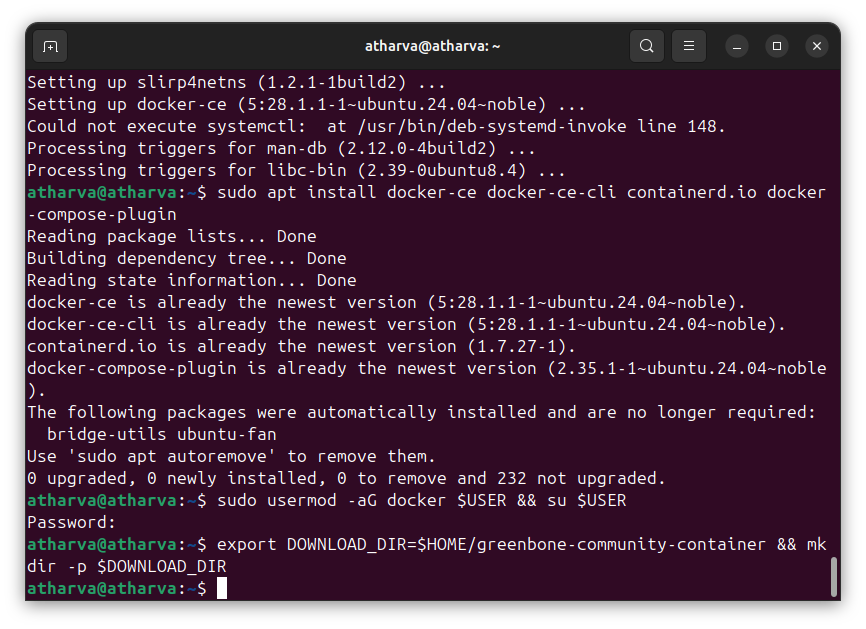
**Greenbone Vulnerability Management (GVM)**

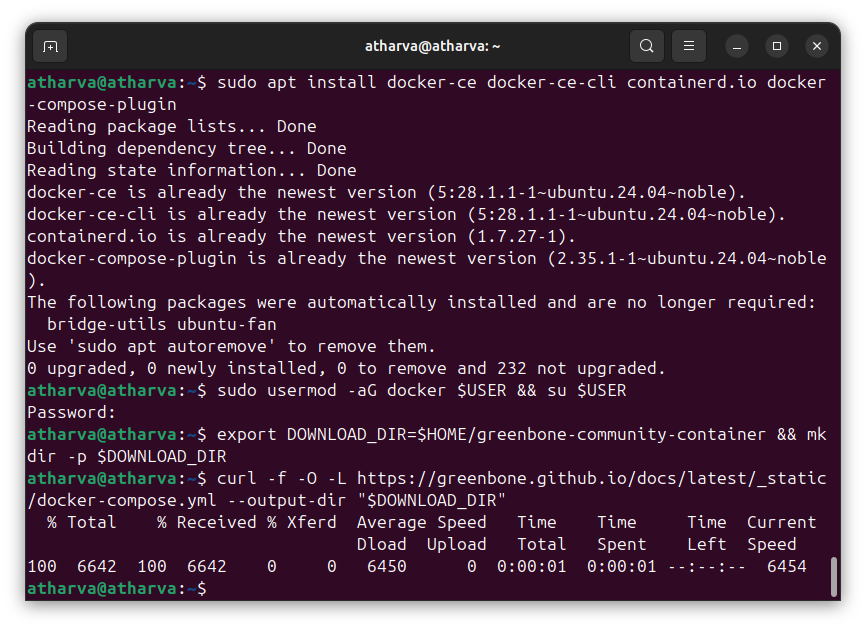
**Greenbone Security Assistant (GSA)**

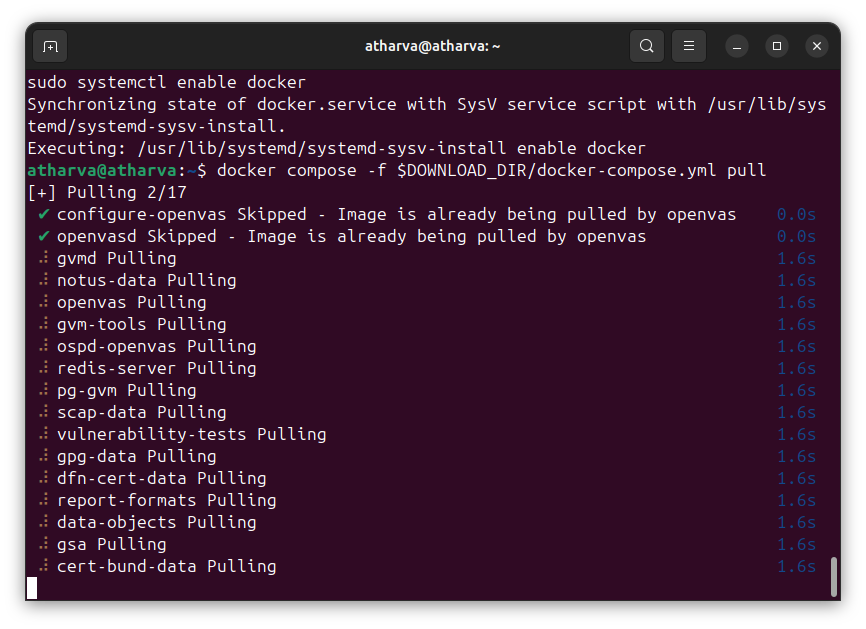
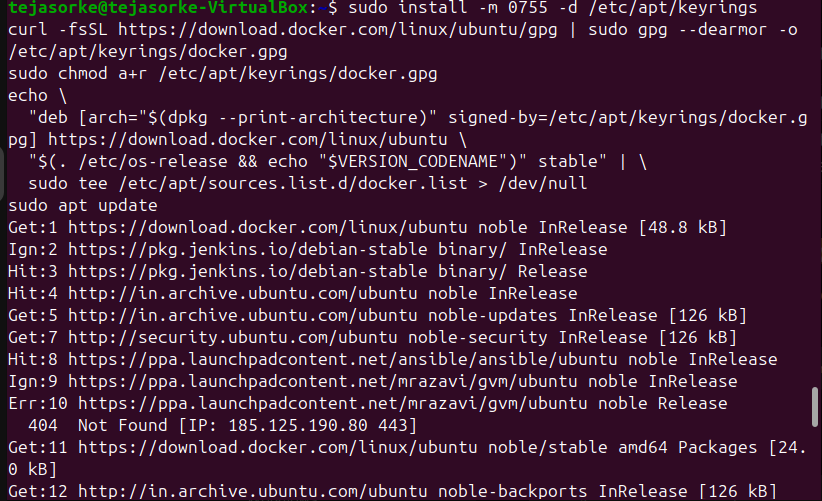


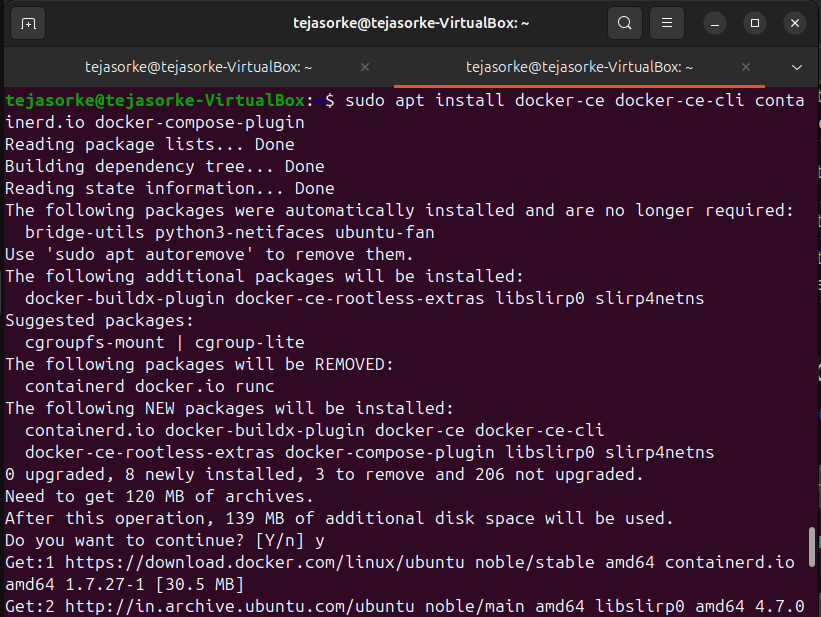












Docker compose.yml

name: greenbone-community-edition

services:

vulnerability-tests:

image: registry.community.greenbone.net/community/vulnerability-tests

environment:

FEED\_RELEASE: "24.10"

volumes:

- vt\_data\_vol:/mnt

notus-data:

image: registry.community.greenbone.net/community/notus-data

volumes:

- notus\_data\_vol:/mnt

scap-data:

image: registry.community.greenbone.net/community/scap-data

volumes:

- scap\_data\_vol:/mnt

cert-bund-data:

image: registry.community.greenbone.net/community/cert-bund-data

volumes:

- cert\_data\_vol:/mnt

dfn-cert-data:

image: registry.community.greenbone.net/community/dfn-cert-data

volumes:

- cert\_data\_vol:/mnt

depends\_on:

- cert-bund-data

data-objects:

image: registry.community.greenbone.net/community/data-objects

environment:

FEED\_RELEASE: "24.10"

volumes:

- data\_objects\_vol:/mnt

report-formats:

image: registry.community.greenbone.net/community/report-formats

environment:

FEED\_RELEASE: "24.10"

volumes:

- data\_objects\_vol:/mnt

depends\_on:

- data-objects

gpg-data:

image: registry.community.greenbone.net/community/gpg-data

volumes:

- gpg\_data\_vol:/mnt

redis-server:

image: registry.community.greenbone.net/community/redis-server

restart: on-failure

volumes:

- redis\_socket\_vol:/run/redis/

pg-gvm:

image: registry.community.greenbone.net/community/pg-gvm:stable

restart: on-failure

volumes:

- psql\_data\_vol:/var/lib/postgresql

- psql\_socket\_vol:/var/run/postgresql

gvmd:

image: registry.community.greenbone.net/community/gvmd:stable

restart: on-failure

volumes:

- gvmd\_data\_vol:/var/lib/gvm

- scap\_data\_vol:/var/lib/gvm/scap-data/

- cert\_data\_vol:/var/lib/gvm/cert-data

- data\_objects\_vol:/var/lib/gvm/data-objects/gvmd

- vt\_data\_vol:/var/lib/openvas/plugins

- psql\_data\_vol:/var/lib/postgresql

- gvmd\_socket\_vol:/run/gvmd

- ospd\_openvas\_socket\_vol:/run/ospd

- psql\_socket\_vol:/var/run/postgresql

depends\_on:

pg-gvm:

condition: service\_started

scap-data:

condition: service\_completed\_successfully

cert-bund-data:

condition: service\_completed\_successfully

dfn-cert-data:

condition: service\_completed\_successfully

data-objects:

condition: service\_completed\_successfully

report-formats:

condition: service\_completed\_successfully

gsa:

image: registry.community.greenbone.net/community/gsa:stable

restart: on-failure

ports:

- 127.0.0.1:9392:80

volumes:

- gvmd\_socket\_vol:/run/gvmd

depends\_on:

- gvmd

# Sets log level of openvas to the set LOG\_LEVEL within the env

# and changes log output to /var/log/openvas instead /var/log/gvm

# to reduce likelyhood of unwanted log interferences

configure-openvas:

image: registry.community.greenbone.net/community/openvas-scanner:stable

volumes:

- openvas\_data\_vol:/mnt

- openvas\_log\_data\_vol:/var/log/openvas

command:

- /bin/sh

- -c

- |

printf "table\_driven\_lsc = yes\nopenvasd\_server = http://openvasd:80\n" > /mnt/openvas.conf

sed "s/127/128/" /etc/openvas/openvas\_log.conf | sed 's/gvm/openvas/' > /mnt/openvas\_log.conf

chmod 644 /mnt/openvas.conf

chmod 644 /mnt/openvas\_log.conf

touch /var/log/openvas/openvas.log

chmod 666 /var/log/openvas/openvas.log

# shows logs of openvas

openvas:

image: registry.community.greenbone.net/community/openvas-scanner:stable

restart: on-failure

volumes:

- openvas\_data\_vol:/etc/openvas

- openvas\_log\_data\_vol:/var/log/openvas

command:

- /bin/sh

- -c

- |

cat /etc/openvas/openvas.conf

tail -f /var/log/openvas/openvas.log

depends\_on:

configure-openvas:

condition: service\_completed\_successfully

openvasd:

image: registry.community.greenbone.net/community/openvas-scanner:stable

restart: on-failure

environment:

# `service\_notus` is set to disable everything but notus,

# if you want to utilize openvasd directly, remove `OPENVASD\_MODE`

OPENVASD\_MODE: service\_notus

GNUPGHOME: /etc/openvas/gnupg

LISTENING: 0.0.0.0:80

volumes:

- openvas\_data\_vol:/etc/openvas

- openvas\_log\_data\_vol:/var/log/openvas

- gpg\_data\_vol:/etc/openvas/gnupg

- notus\_data\_vol:/var/lib/notus

# enable port forwarding when you want to use the http api from your host machine

# ports:

# - 127.0.0.1:3000:80

depends\_on:

vulnerability-tests:

condition: service\_completed\_successfully

configure-openvas:

condition: service\_completed\_successfully

gpg-data:

condition: service\_completed\_successfully

networks:

default:

aliases:

- openvasd

ospd-openvas:

image: registry.community.greenbone.net/community/ospd-openvas:stable

restart: on-failure

hostname: ospd-openvas.local

cap\_add:

- NET\_ADMIN # for capturing packages in promiscuous mode

- NET\_RAW # for raw sockets e.g. used for the boreas alive detection

security\_opt:

- seccomp=unconfined

- apparmor=unconfined

command:

[

"ospd-openvas",

"-f",

"--config",

"/etc/gvm/ospd-openvas.conf",

"--notus-feed-dir",

"/var/lib/notus/advisories",

"-m",

"666",

]

volumes:

- gpg\_data\_vol:/etc/openvas/gnupg

- vt\_data\_vol:/var/lib/openvas/plugins

- notus\_data\_vol:/var/lib/notus

- ospd\_openvas\_socket\_vol:/run/ospd

- redis\_socket\_vol:/run/redis/

- openvas\_data\_vol:/etc/openvas/

- openvas\_log\_data\_vol:/var/log/openvas

depends\_on:

redis-server:

condition: service\_started

gpg-data:

condition: service\_completed\_successfully

vulnerability-tests:

condition: service\_completed\_successfully

configure-openvas:

condition: service\_completed\_successfully

gvm-tools:

image: registry.community.greenbone.net/community/gvm-tools

volumes:

- gvmd\_socket\_vol:/run/gvmd

- ospd\_openvas\_socket\_vol:/run/ospd

depends\_on:

- gvmd

- ospd-openvas

volumes:

gpg\_data\_vol:

scap\_data\_vol:

cert\_data\_vol:

data\_objects\_vol:

gvmd\_data\_vol:

psql\_data\_vol:

vt\_data\_vol:

notus\_data\_vol:

psql\_socket\_vol:

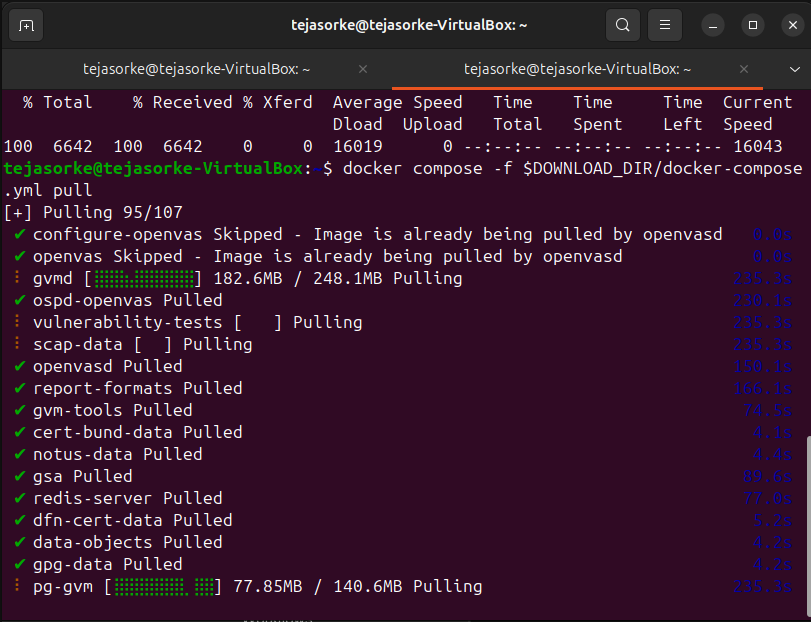
gvmd\_socket\_vol:

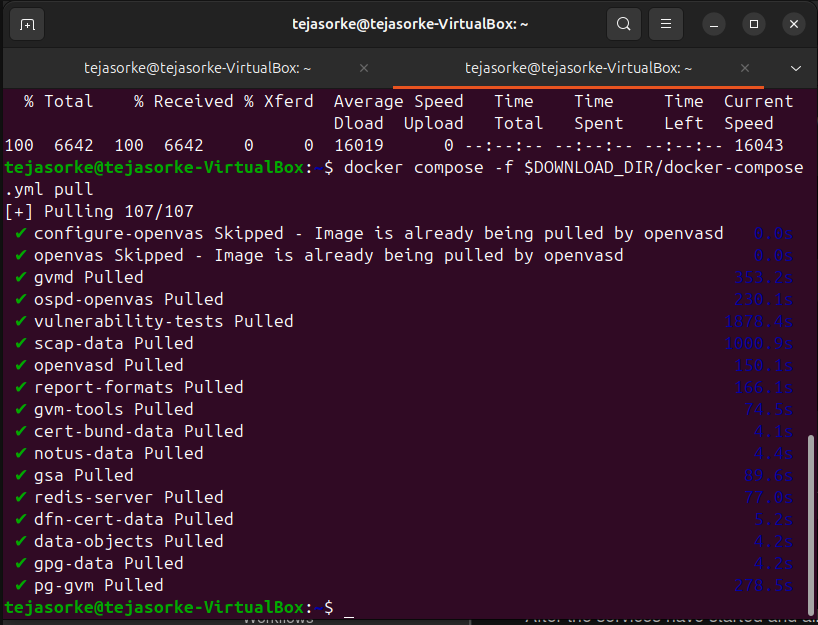
ospd\_openvas\_socket\_vol:

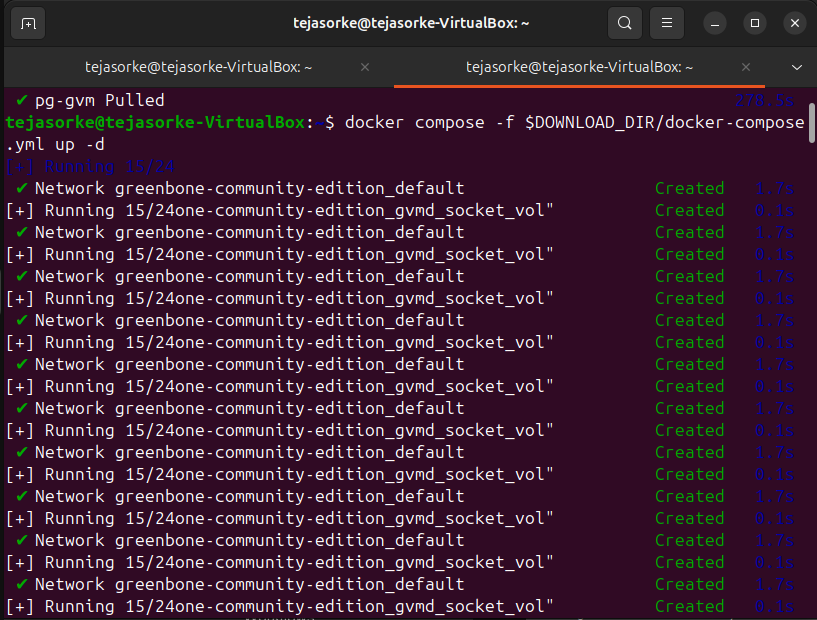
redis\_socket\_vol:

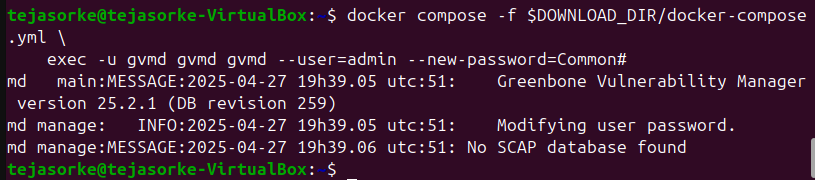
openvas\_data\_vol:

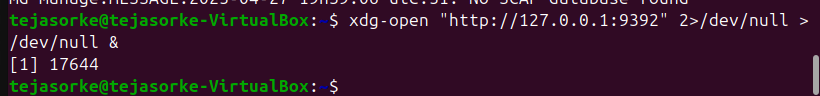
openvas\_log\_data\_vol:

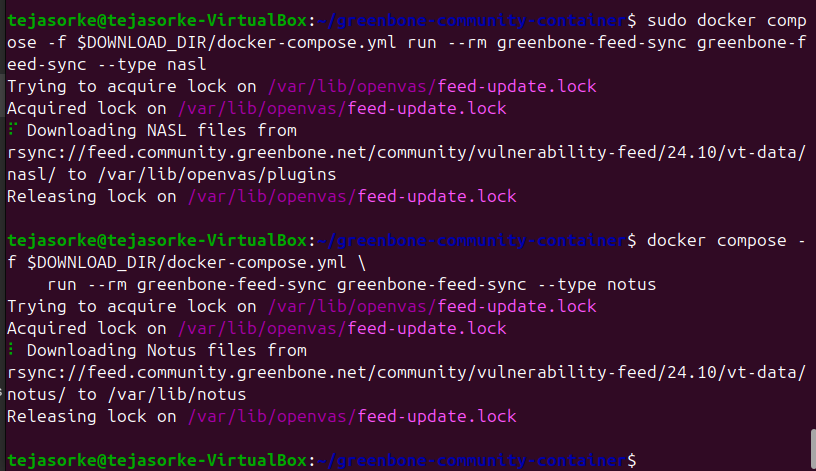


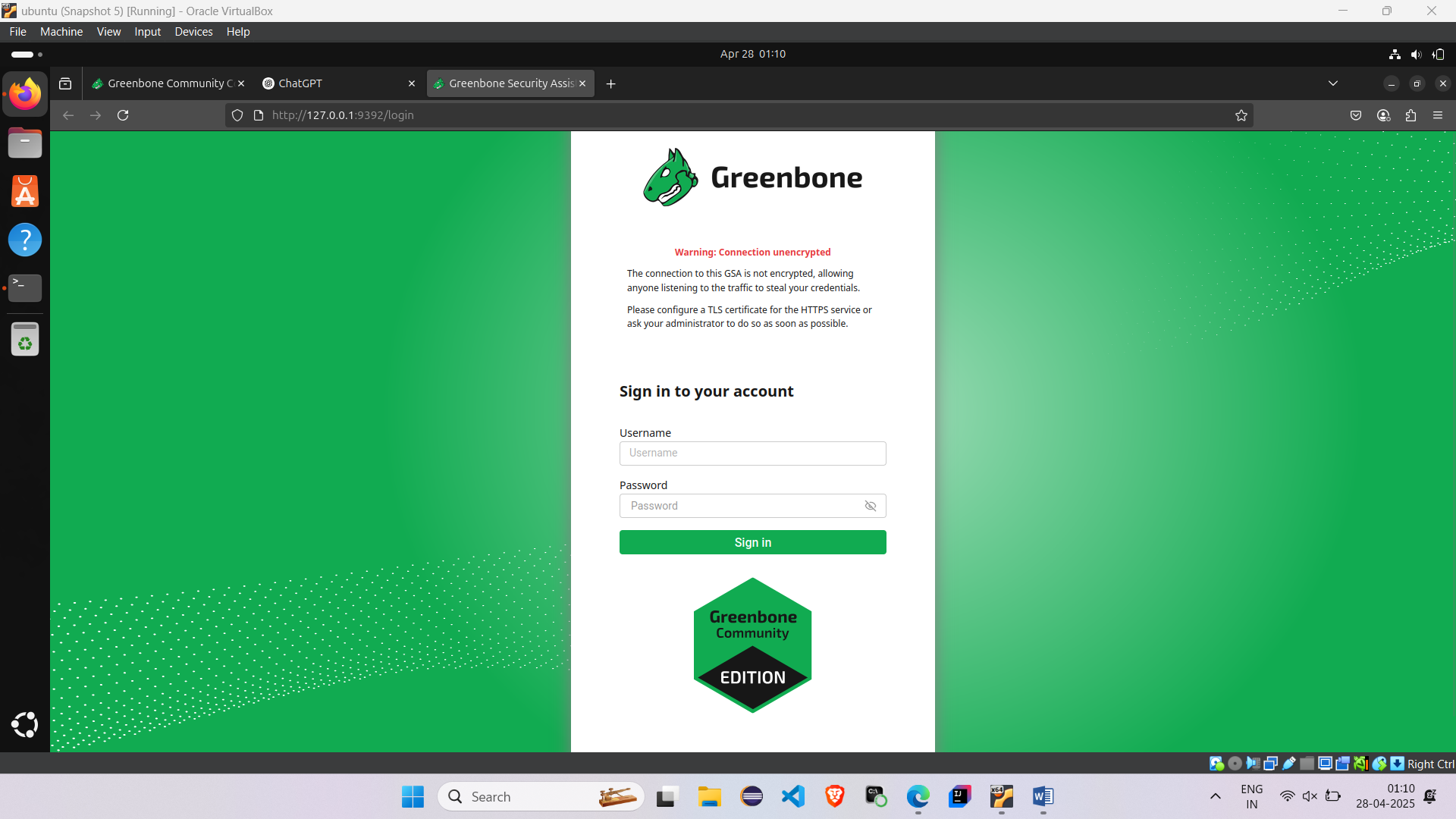


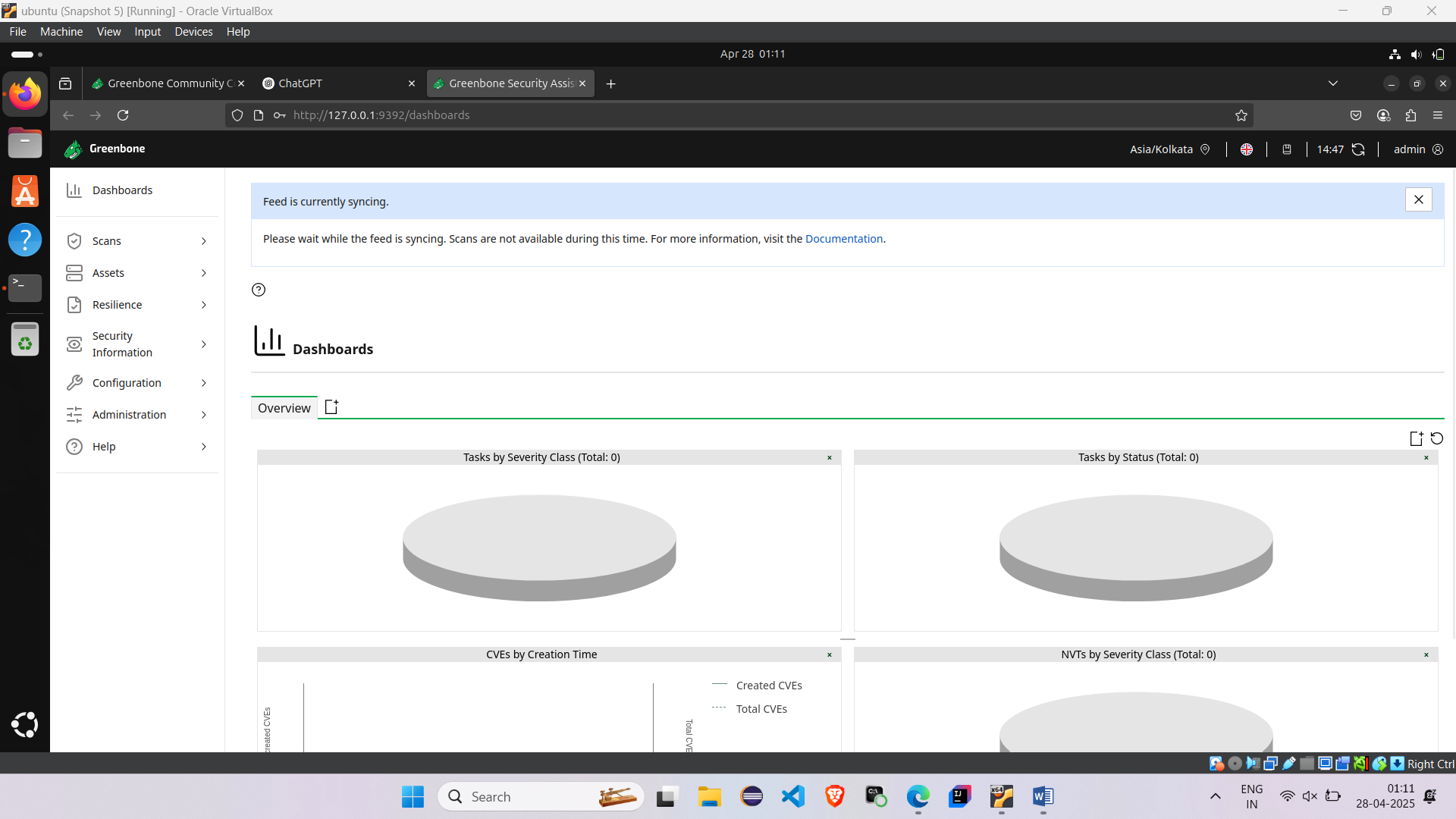


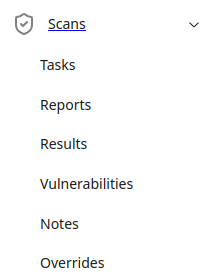


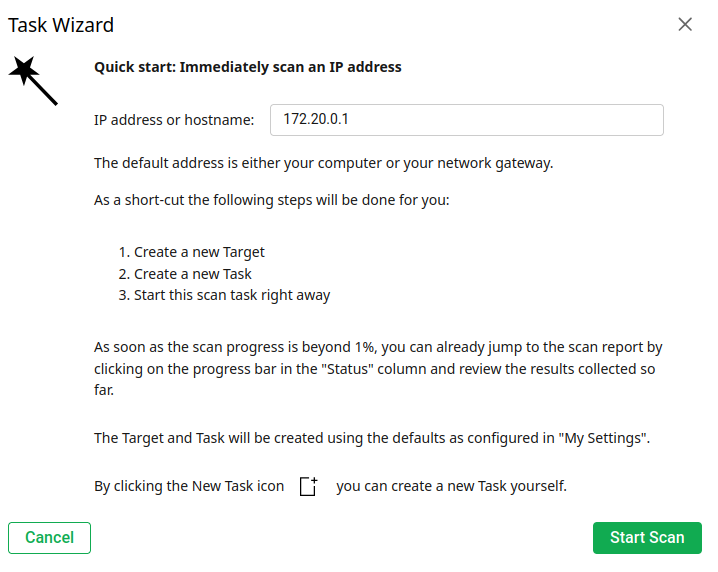


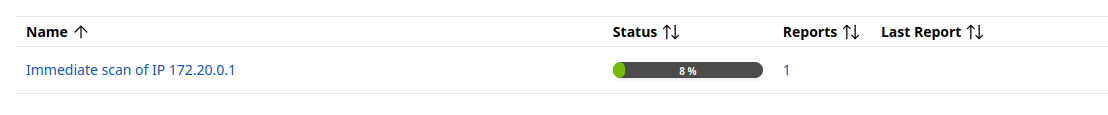












**Observation :**

The Docker-based deployment of Greenbone Vulnerability Management (GVM) using the provided docker-compose.yml demonstrates a modular and efficient setup for proactive cybersecurity monitoring. Each component—such as vulnerability feeds, OpenVAS scanner, and Greenbone Security Assistant (GSA)—is containerized for better scalability and isolation. The configuration supports real-time vulnerability detection, feed synchronization, and detailed report generation through GSA's web interface. By using volume mapping and service dependencies, the system ensures data persistence and smooth inter-container communication. Overall, this setup provides a robust foundation for automated, continuous vulnerability assessment in a networked environment.