Lab - Installing OpenVAS Using Docker

Overview

In this lab, you will install the docker program onto your Kali machine. Once that has been completed, you will move on to the second lab and download and install the Docker container for OpenVAS.

Using Docker, we will be able to install OpenVAS and all its dependencies without having to install or use any of the dependencies on our Kali machine.

About docker

Docker is a platform for developers and sysadmins to develop, deploy, and run applications with containers. The use of Linux containers to deploy applications is called containerization. Containers are not new, but their use for easily deploying applications is.

Containerization is increasingly popular because containers are:

- Flexible: Even the most complex applications can be containerized.
- Lightweight: Containers leverage and share the host kernel.
- Interchangeable: You can deploy updates and upgrades on-the-fly.
- Portable: You can build locally, deploy to the cloud, and run anywhere.
- Scalable: You can increase and automatically distribute container replicas.
- Stackable: You can stack services vertically and on-the-fly.
- Containers are portable

Images and containers

A container is launched by running an image. An image is an executable package that includes everything needed to run an application--the code, a runtime, libraries, environment variables, and configuration files.

A container is a runtime instance of an image--what the image becomes in memory when executed (that is, an image with state, or a user process). You can see a list of your running containers with the command, **`docker ps`**, just as you would in Linux.

Containers and virtual machines

A container runs natively on Linux and shares the kernel of the host machine with other containers. It runs a discrete process, taking no more memory than any other executable, making it lightweight.

By contrast, a virtual machine (VM) runs a full-blown “guest” operating system with virtual access to host resources through a hypervisor. In general, VMs provide an environment with more resources than most applications need.

Reference:

https://docs.docker.com/get-started/

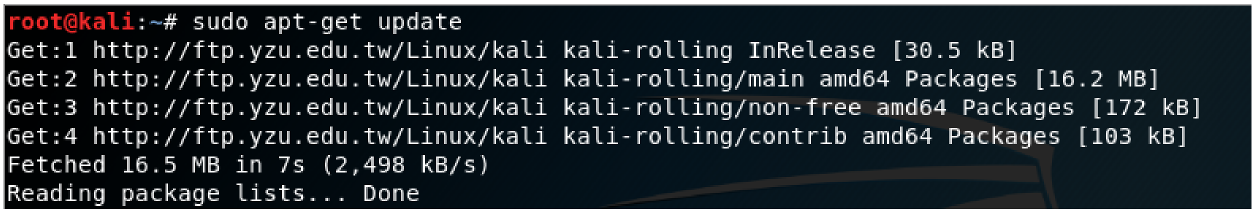
Requirements

- One virtual install of Kali Linux.
- Kali has been recently updated and upgraded with the latest packages.
- Internet connection

Begin the lab

Ensure Kali has been updated.

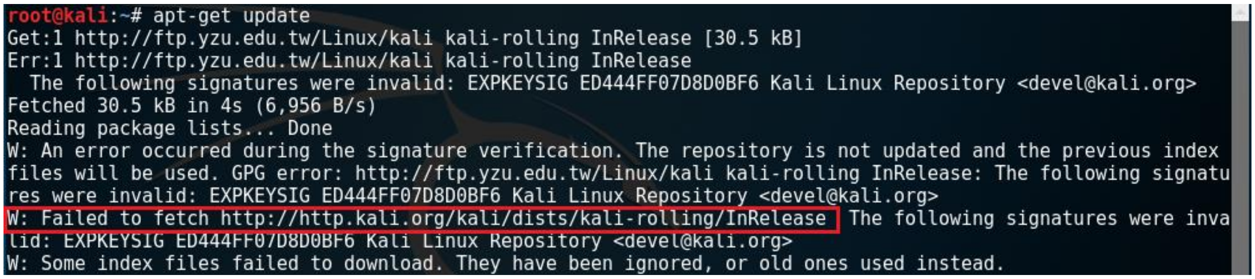
```
apt-get update
```



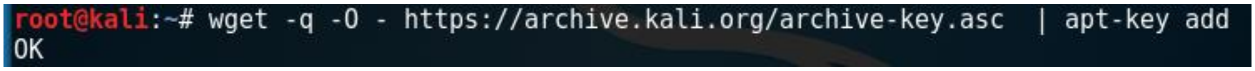
```
root@kali:~# sudo apt-get update
Get:1 http://ftp.yzu.edu.tw/Linux/kali kali-rolling InRelease [30.5 kB]
Get:2 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 Packages [16.2 MB]
Get:3 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/non-free amd64 Packages [172 kB]
Get:4 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/contrib amd64 Packages [103 kB]
Fetched 16.5 MB in 7s (2,498 kB/s)
Reading package lists... Done
```

If you get an error referencing an invalid key signature, you need to update your key signature using the following command:

```
wget -q -O - https://archive.kali.org/archive-key.asc  | apt-key add
```

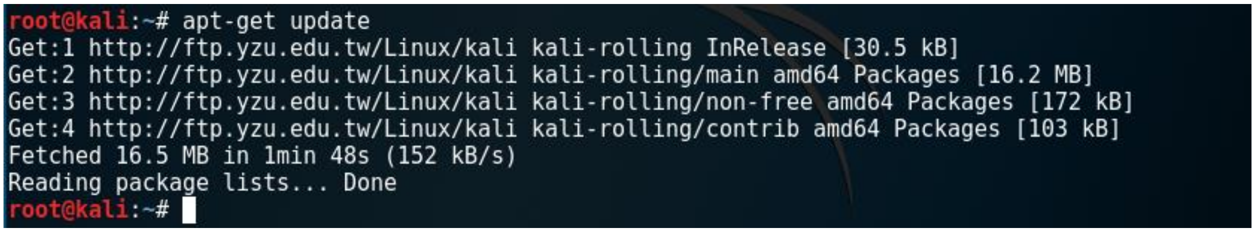


```
root@kali:~# apt-get update
Get:1 http://ftp.yzu.edu.tw/Linux/kali kali-rolling InRelease [30.5 kB]
Err:1 http://ftp.yzu.edu.tw/Linux/kali kali-rolling InRelease
  The following signatures were invalid: EXPKEYSIG ED444FF07D8D0BF6 Kali Linux Repository <devel@kali.org>
Fetched 30.5 kB in 4s (6,956 B/s)
Reading package lists... Done
W: An error occurred during the signature verification. The repository is not updated and the previous index
files will be used. GPG error: http://ftp.yzu.edu.tw/Linux/kali kali-rolling InRelease: The following signatu
res were invalid: EXPKEYSIG ED444FF07D8D0BF6 Kali Linux Repository <devel@kali.org>
W: Failed to fetch http://http.kali.org/kali/dists/kali-rolling/InRelease  The following signatures were inva
lid: EXPKEYSIG ED444FF07D8D0BF6 Kali Linux Repository <devel@kali.org>
W: Some index files failed to download. They have been ignored, or old ones used instead.
```



```
root@kali:~# wget -q -O - https://archive.kali.org/archive-key.asc  | apt-key add
OK
```

Run the update command once again.



```
root@kali:~# apt-get update
Get:1 http://ftp.yzu.edu.tw/Linux/kali kali-rolling InRelease [30.5 kB]
Get:2 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/main amd64 Packages [16.2 MB]
Get:3 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/non-free amd64 Packages [172 kB]
Get:4 http://ftp.yzu.edu.tw/Linux/kali kali-rolling/contrib amd64 Packages [103 kB]
Fetched 16.5 MB in 1min 48s (152 kB/s)
Reading package lists... Done
root@kali:~# 
```

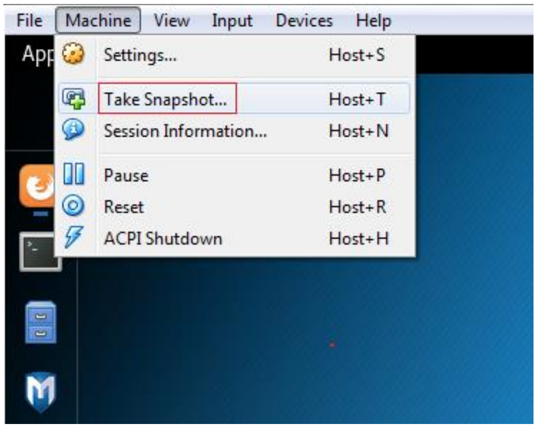
Once you're done with the apt-get update, continue with the apt-get upgrade command.

Creating a Snapshot of Your Current Kali Configuration

Before making any changes to your Kali install, you can take a snapshot of the current configuration so that if needed, you can rollback you, Kali, before the changes were made.

For VirtualBox, with your Kali running, from the VirtualBox taskbar, click on the machine and from the context menu, select, Take Snapshot.

To create a snapshot using VMWare, you will need the Workstation Pro version. Creating a snapshot using the VMWare Free Player is not an option.



If you previously installed the Docker Program for the installation of NESSUS, no need to install it a second time. Click the link to advance to the Download and Install the OpenVAS Docker Container section of the lab.

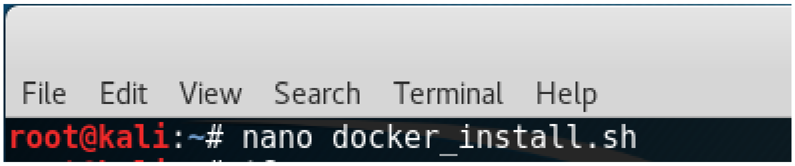
Installing the Docker Program

To install the Docker program, we create a script that will automate the entire process. To build the script, we can use any text editor Kali provides. For this demonstration, we will be using Nano, but you are free to use the text editor of your choice.

Building the Docker Installation Install Script

Once Kali has been updated, and your Snapshot has been completed, at the terminal, type the following:

```
nano docker_install.sh
```

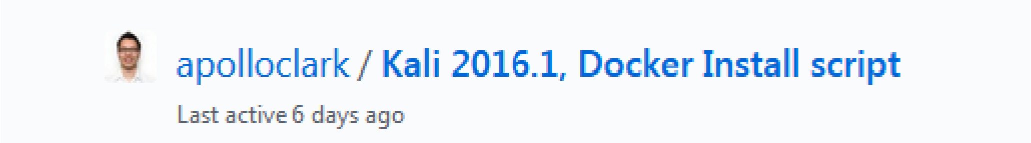


```
root@kali:~# nano docker_install.sh
```

This opens a blank text file using the nano text editor.

The script we will be using is available at https://gist.github.com/apolloclark/f0e3974601346883c731

Thanks to appolloclark for creating and sharing this script!



apolloclark / **Kali 2016.1, Docker Install script**

Last active 6 days ago

Copy and paste the following text between the lines into the blank text editor.

Copy Only Text Inside the Box

```
#!/bin/bash

# update apt-get
export DEBIAN_FRONTEND="noninteractive"
sudo apt-get update

# remove previously installed Docker
sudo apt-get purge lxc-docker*
sudo apt-get purge docker.io*

# add Docker repo
sudo apt-get install -y apt-transport-https ca-certificates
sudo apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-
keys 58118E89F3A912897C070ADBF76221572C52609D

cat > /etc/apt/sources.list.d/docker.list <<'EOF'
deb https://apt.dockerproject.org/repo debian-stretch main
EOF
sudo apt-get update

# install Docker
sudo apt-get install -y docker-engine
sudo service docker start
sudo docker run hello-world


# configure Docker user group permissions
sudo groupadd docker
sudo gpasswd -a ${USER} docker
sudo service docker restart

# set Docker to auto-launch on startup
sudo systemctl enable docker
```

Save the Script

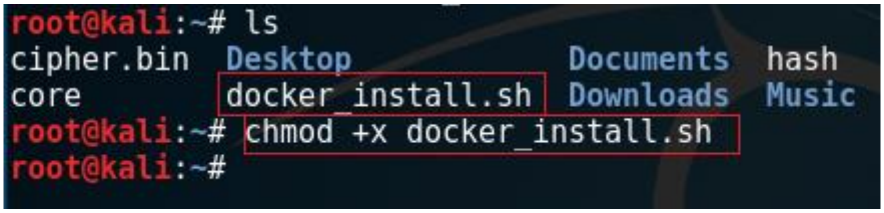
Save the file by pressing CTRL+x.

Type in 'y' to save the changes and then press enter to exit.

At the terminal, type **`ls`** to see the location your newly created script file.

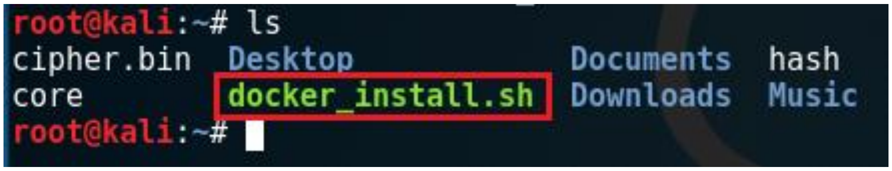
Type the following to make the script executable:

```
chmod +x docker_install.sh
```



```
root@kali:~# ls
cipher.bin  Desktop            Documents  hash
core        docker_install.sh  Downloads  Music
root@kali:~# chmod +x docker_install.sh
root@kali:~#
```

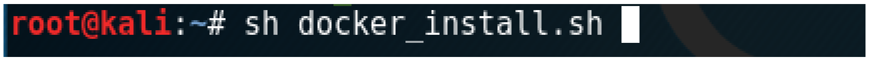
Type in `ls` and note the color of the file has changed to green annotating that the file is now an executable.



```
root@kali:~# ls
cipher.bin  Desktop            Documents  hash
core        docker_install.sh  Downloads  Music
root@kali:~# 
```

To run the script, at the terminal type:

```
Sh docker_install.sh
```



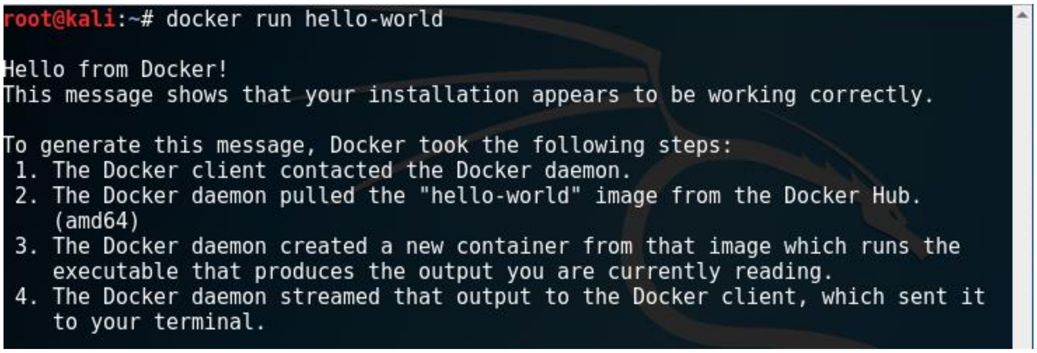
```
root@kali:~# sh docker_install.sh 
```

Hit enter

Allow the script to run and do not interrupt!

Check to see if Docker is properly installed

```
docker run hello-world
```

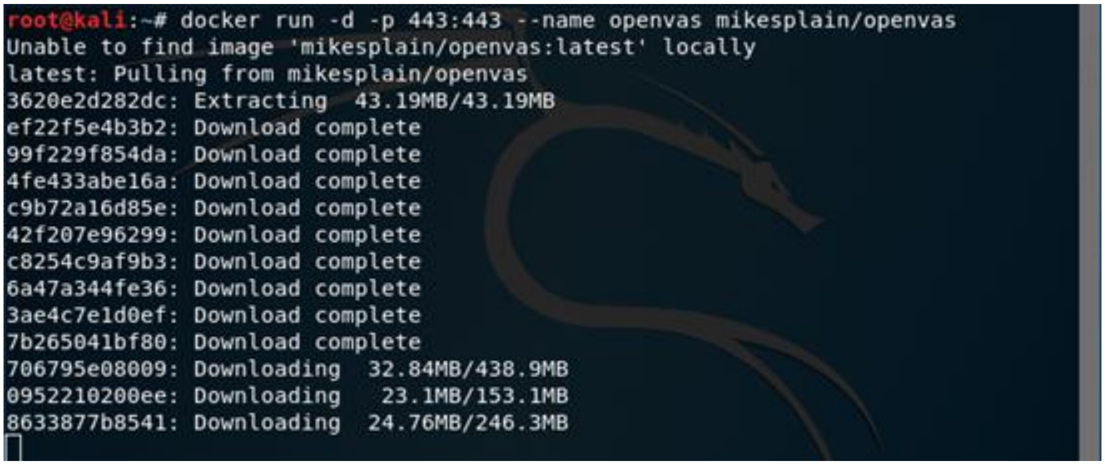


```
root@kali:~# docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

Download and Install the OpenVAS Docker Container

Open a new terminal in Kali Linux. Run the following command.

```
docker run -d -p 443:443 --name openvas mikesplain/openvas
```



```
root@kali:~# docker run -d -p 443:443 --name openvas mikesplain/openvas
Unable to find image 'mikesplain/openvas:latest' locally
latest: Pulling from mikesplain/openvas
3620e2d282dc: Extracting  43.19MB/43.19MB
ef22f5e4b3b2: Download complete
99f229f854da: Download complete
4fe433abe16a: Download complete
c9b72a16d85e: Download complete
42f207e96299: Download complete
c8254c9af9b3: Download complete
6a47a344fe36: Download complete
3ae4c7e1d0ef: Download complete
7b265041bf80: Download complete
706795e08009: Downloading  32.84MB/438.9MB
0952210200ee: Downloading   23.1MB/153.1MB
8633877b8541: Downloading  24.76MB/246.3MB
```

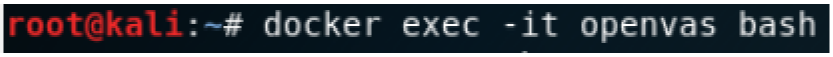
Updating the OpenVAS NVT's

Greenbone maintains a public feed of **Network Vulnerability Tests (NVTs)** for the OpenVAS project, the Greenbone Community Feed. It contains more than 50,000 NVTs, growing on a permanent basis. This feed is configured as the default for the OpenVAS Scanner and relates to the Greenbone Security Feed which is part of the commercial Greenbone Security Manager appliance products.

Occasionally you'll need to update NVTs. You can update your container by running the following commands:

Open a terminal and type the following command.

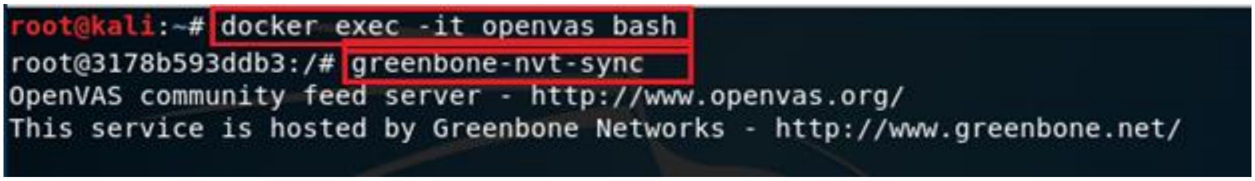
`docker exec -it openvas bash` (This starts an interactive BASH prompt within the OpenVAS container)



```
root@kali:~# docker exec -it openvas bash
```

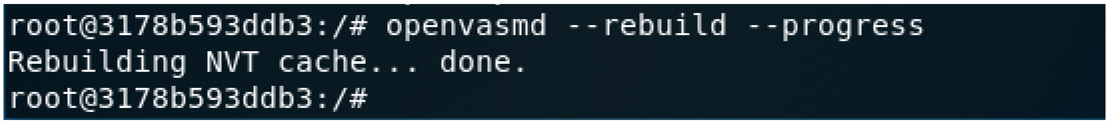
At the prompt copy and paste all the following commands, one at a time.

```
greenbone-nvt-sync
```



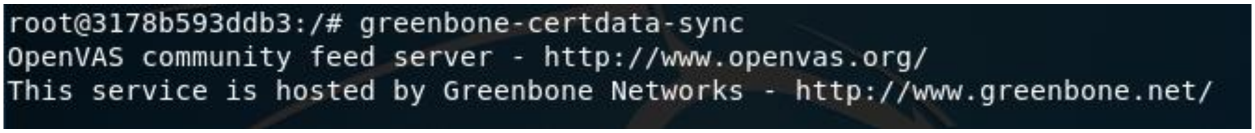
```
root@kali:~# docker exec -it openvas bash
root@3178b593ddb3:/# greenbone-nvt-sync
OpenVAS community feed server - http://www.openvas.org/
This service is hosted by Greenbone Networks - http://www.greenbone.net/
```

`openvasmd --rebuild --progress` (This will take some time to complete, just watch the prompt)



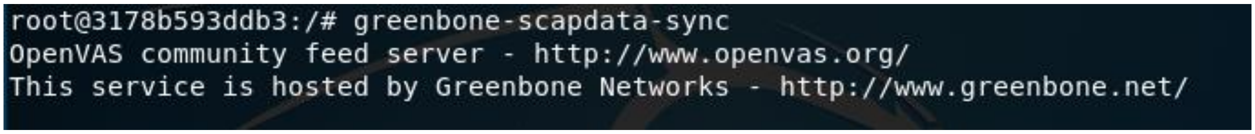
```
root@3178b593ddb3:/# openvasmd --rebuild --progress
Rebuilding NVT cache... done.
root@3178b593ddb3:/#
```

```
greenbone-certdata-sync
```



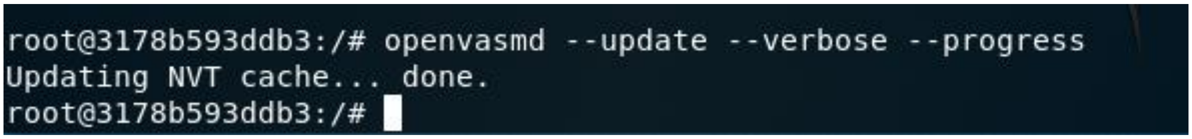
```
root@3178b593ddb3:/# greenbone-certdata-sync
OpenVAS community feed server - http://www.openvas.org/
This service is hosted by Greenbone Networks - http://www.greenbone.net/
```

```
greenbone-scapdata-sync
```



```
root@3178b593ddb3:/# greenbone-scapdata-sync
OpenVAS community feed server - http://www.openvas.org/
This service is hosted by Greenbone Networks - http://www.greenbone.net/
```

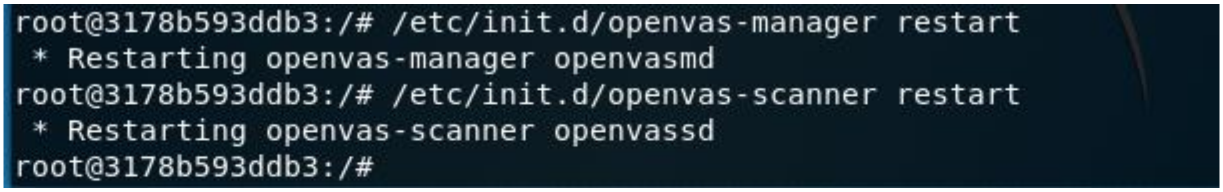
```
openvasmd --update --verbose --progress
```



```
root@3178b593ddb3:/# openvasmd --update --verbose --progress
Updating NVT cache... done.
root@3178b593ddb3:/#
```

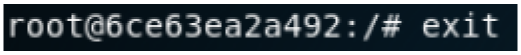
When the updating has completed, restart the scanner and the OpenVAS manager using the following two commands:

```
/etc/init.d/openvas-manager restart
/etc/init.d/openvas-scanner restart
```



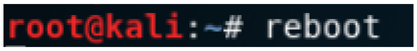
```
root@3178b593ddb3:/# /etc/init.d/openvas-manager restart
 * Restarting openvas-manager openvasmd
root@3178b593ddb3:/# /etc/init.d/openvas-scanner restart
 * Restarting openvas-scanner openvassd
root@3178b593ddb3:/#
```

At the BASH prompt, type, `exit`



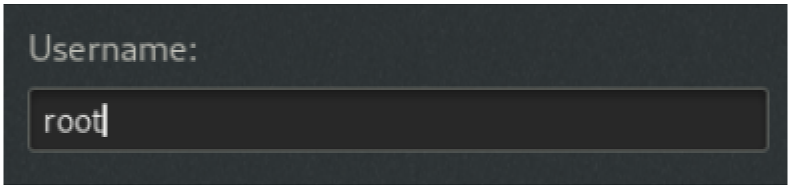
```
root@6ce63ea2a492:/# exit
```

At the Kali prompt, `type, reboot`



```
root@kali:~# reboot
```

`Your Kali machine reboots. Log back in as root`



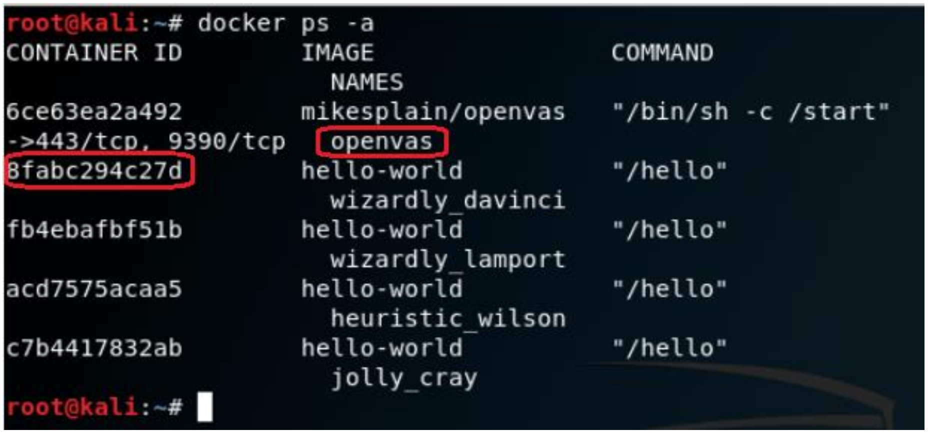
Username:

root

Restarting a Container in Docker

Each time we launch a program using Docker, it creates a new container. For instance, when we ran the **docker run hello-world** command. Docker created a container to run the program.

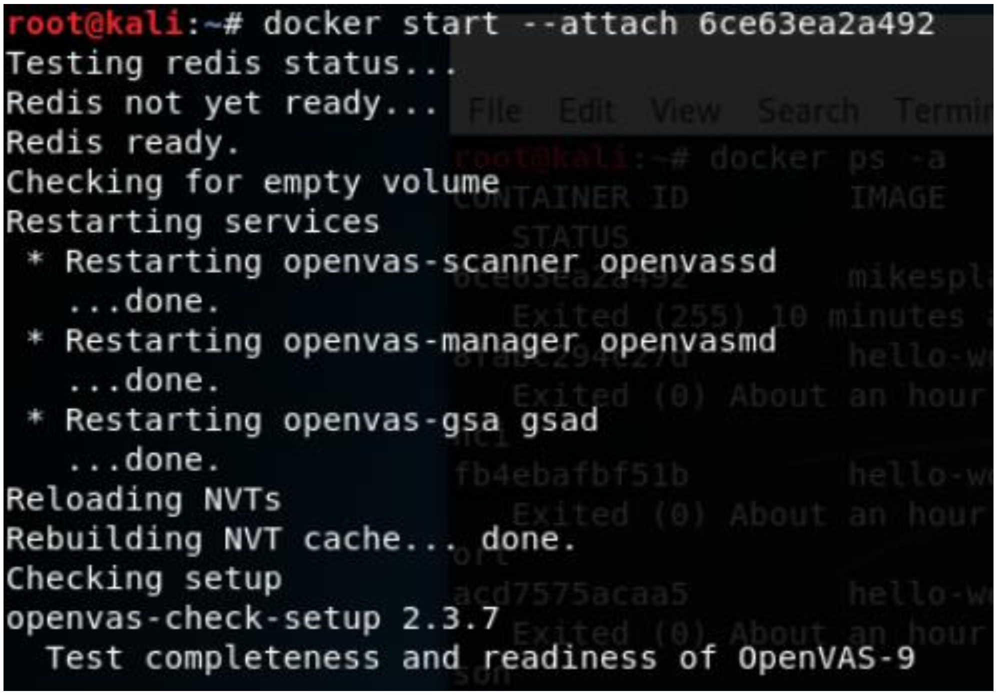
Once the container stops (i.e., when you exit the shell or reboot), it is not deleted by default. You can view all the containers, including stopped ones the typing the following command in the Kali terminal:
`docker ps -a`
You are given the container ID and the name if the program that owns the container.



```
root@kali:~# docker ps -a
CONTAINER ID        IMAGE                  COMMAND
  NAMES
6ce63ea2a492        mikesplain/openvas     "/bin/sh -c /start"
->443/tcp, 9390/tcp   openvas
8fabc294c27d        hello-world            "/hello"
  wizardly_davinci
fb4ebafbf51b        hello-world            "/hello"
  wizardly_lamport
acd7575acaa5        hello-world            "/hello"
  heuristic_wilson
c7b4417832ab        hello-world            "/hello"
  jolly_cray
root@kali:~# 
```

If you need to get back into the specific container, you can use the:

`docker start --attach <CONTAINER ID>`

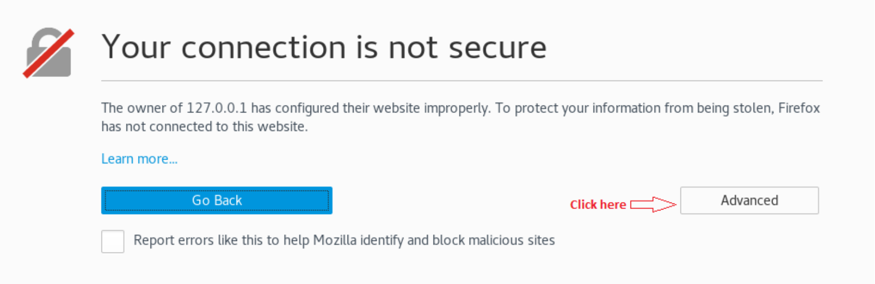


```
root@kali:~# docker start --attach 6ce63ea2a492
Testing redis status...
Redis not yet ready...
Redis ready.
Checking for empty volume
Restarting services
 * Restarting openvas-scanner openvassd
   ...done.
 * Restarting openvas-manager openvasmd
   ...done.
 * Restarting openvas-gsa gsad
   ...done.
Reloading NVTs
Rebuilding NVT cache... done.
Checking setup
openvas-check-setup 2.3.7
  Test completeness and readiness of OpenVAS-9
```

Be patient with the restart of the OpenVAS container. It runs several checks and updates any missing or newer NVTs. Once the log entries have been shown, you will see a blinking cursor with no prompt. You are free to close the terminal. The OpenVAS container is now running.

You can open your Kali browser and type https://127.0.0.1 to get to the GUI manager for OpenVAS.

We need to add an exception for the certificate.

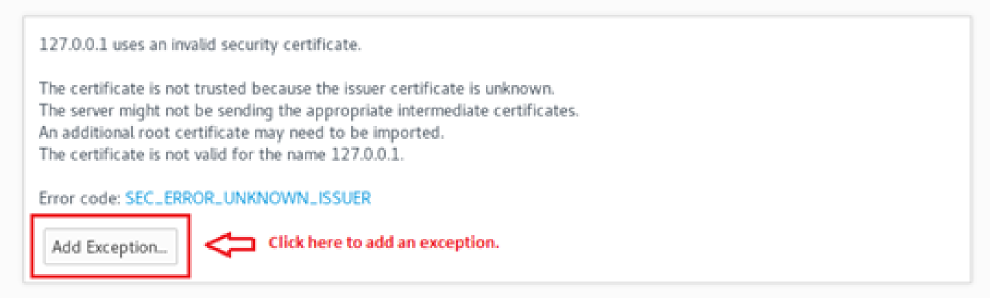


Your connection is not secure

The owner of 127.0.0.1 has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

□ Report errors like this to help Mozilla identify and block malicious sites

Add the exception.

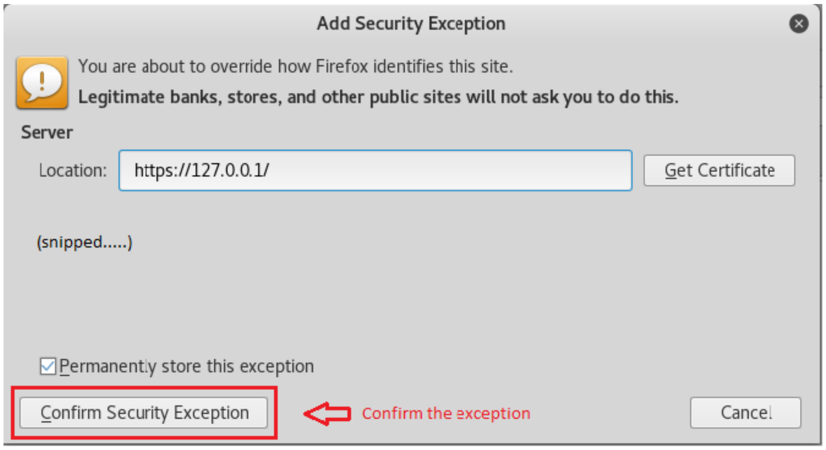


127.0.0.1 uses an invalid security certificate.

The certificate is not trusted because the issuer certificate is unknown.
The server might not be sending the appropriate intermediate certificates.
An additional root certificate may need to be imported.
The certificate is not valid for the name 127.0.0.1.

Error code: SEC_ERROR_UNKNOWN_ISSUER

Confirm the exception.



Add Security Exception

You are about to override how Firefox identifies this site.
Legitimate banks, stores, and other public sites will not ask you to do this.

Server

Location: https://127.0.0.1/

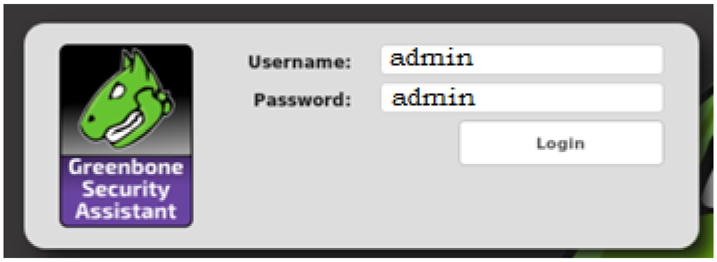
(snipped.....)

☑ Permanently store this exception

Login to OpenVAS.

At the login screen type in the admin credentials for OpenVAS.

```
Username: admin
Password: admin
```



Username: admin
Password: admin

You're now ready to begin a vulnerability scan using OpenVAS. Move on to your next lab, **Lab - Scanning for Vulnerabilities Using OpenVAS.**

Summary

A lot was going on with this lab. We used the Docker program to quickly and effortlessly install an otherwise difficult program, OpenVAS. We were able to install the program without making any changes to the Kali operating system and update the program inside its container. If you are not comfortable with using Docker, you need to get there. This is not going away. Learning new technologies is what we do. There are plenty of techs that can work on AS400s and Apple Servers but there isn't a huge demand for either.