Lab 11 Security monitoring

Name: Ahmed Baga Eddine Alimi
Date: April 14, 2025
Class: SD-01

Exercise 1: Set up Wazuh

Install the Wazuh server on your host machine

To start we need to setup **Wazuh** on the ubuntu machine we are going to use the quick setup payload. It includes the installation of required packages and preparation of the environment for running the Wazuh server.

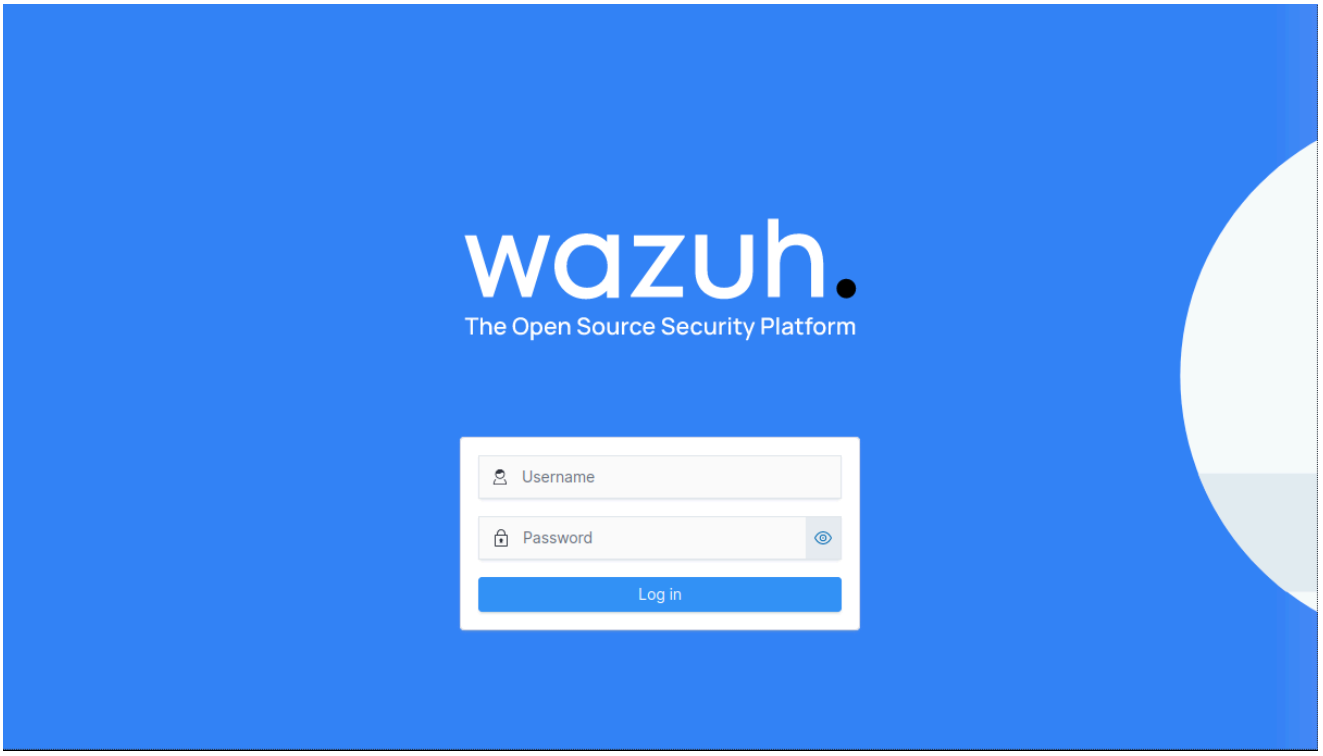
```
wazuh@wazuh:~$ curl -sO https://packages.wazuh.com/4.11/wazuh-install.sh && sudo bash ./wazuh-install.sh -a
14/04/2025 21:59:19 INFO: Starting Wazuh installation assistant. Wazuh version: 4.11.2 (x86_64/AMD64)
14/04/2025 21:59:19 INFO: Verbose logging redirected to /var/log/wazuh-install.log
14/04/2025 22:01:23 INFO: --- Dependencies ----
14/04/2025 22:01:23 INFO: Installing gawk.
14/04/2025 22:01:39 INFO: Verifying that your system meets the recommended minimum hardware requirements.
14/04/2025 22:01:39 INFO: Wazuh web interface port will be 443.
14/04/2025 22:01:47 INFO: --- Dependencies ----
14/04/2025 22:01:47 INFO: Installing apt-transport-https.
14/04/2025 22:01:55 INFO: Installing debhelper.
14/04/2025 22:14:46 INFO: Wazuh repository added.
14/04/2025 22:14:46 INFO: --- Configuration files ---
14/04/2025 22:14:46 INFO: Generating configuration files.
14/04/2025 22:14:47 INFO: Generating the root certificate.
14/04/2025 22:14:47 INFO: Generating Admin certificates.
14/04/2025 22:14:47 INFO: Generating Wazuh indexer certificates.
14/04/2025 22:14:47 INFO: Generating Filebeat certificates.
14/04/2025 22:14:47 INFO: Generating Wazuh dashboard certificates.
14/04/2025 22:14:48 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary
for installation.
14/04/2025 22:14:48 INFO: --- Wazuh indexer ---
14/04/2025 22:14:48 INFO: Starting Wazuh indexer installation.
```

```
    User: admin
    Password: AtYUFa?os7qQcz00oWDk.0LjxJn1Eiy4
14/04/2025 22:53:31 INFO: --- Dependencies ----
14/04/2025 22:53:31 INFO: Removing gawk.
14/04/2025 22:53:40 INFO: Installation finished.
wazuh@wazuh:~$
```

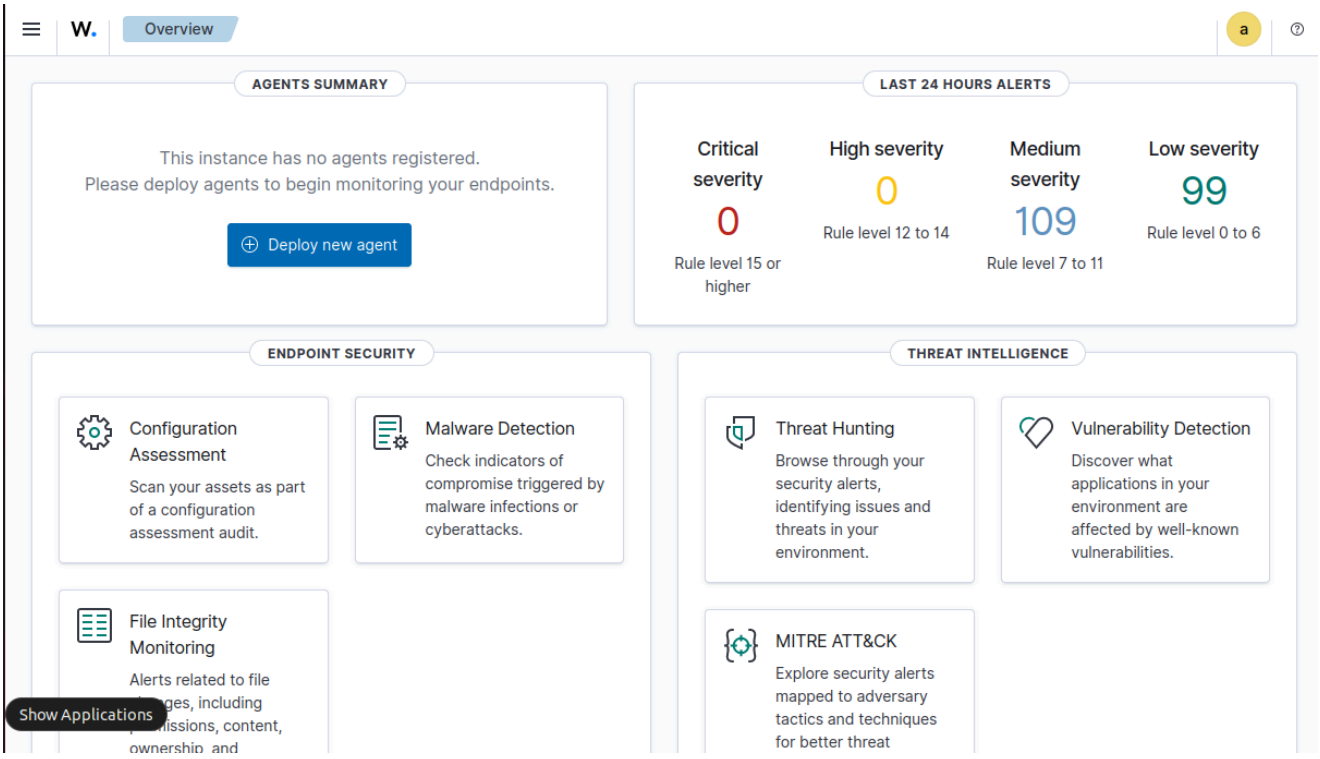
At the end of the Wazuh installation process, the terminal output displays the default credentials required to access the web interface. The username is set to `admin`, and a randomly generated secure password is provided. Once the installation is complete, the script automatically removes temporary dependencies such as `gawk`, confirming a clean and successful setup. The Wazuh server is now fully operational and accessible via HTTPS on port 443.

```
wazuh@wazuh:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:1a:df:45 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.160.129/24 brd 192.168.160.255 scope global dynamic noprefixroute ens33
       valid_lft 1078sec preferred_lft 1078sec
    inet6 fe80::f30f:4bda:438c:267f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

To determine the IP address of the web interface, the `ip addr` command is used. This allows us to identify the local IP assigned to the server. In this case, the IP address is `192.168.160.129`. This is the address that we will use to access the Wazuh web interface over HTTPS on port 443.

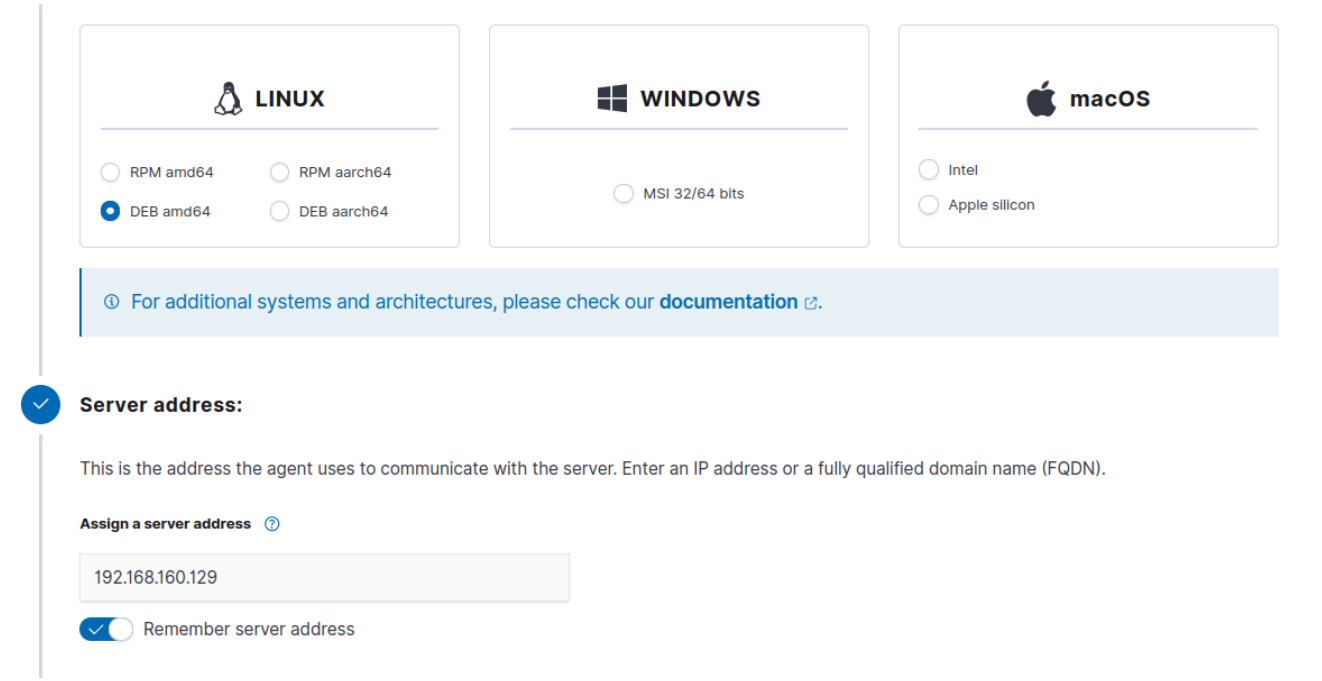


The login page presented a clean authentication form where I entered the admin credentials created during setup. Upon successful login, the dashboard provided immediate visibility into security events, agent status, and threat detection features



Install the Wazuh agent on a second Linux machine

Wazuh offers a simple agent deployment method. Just install the appropriate package (Linux RPM/DEB or Windows MSI) and configure it with the server IP (192.168.160.129)

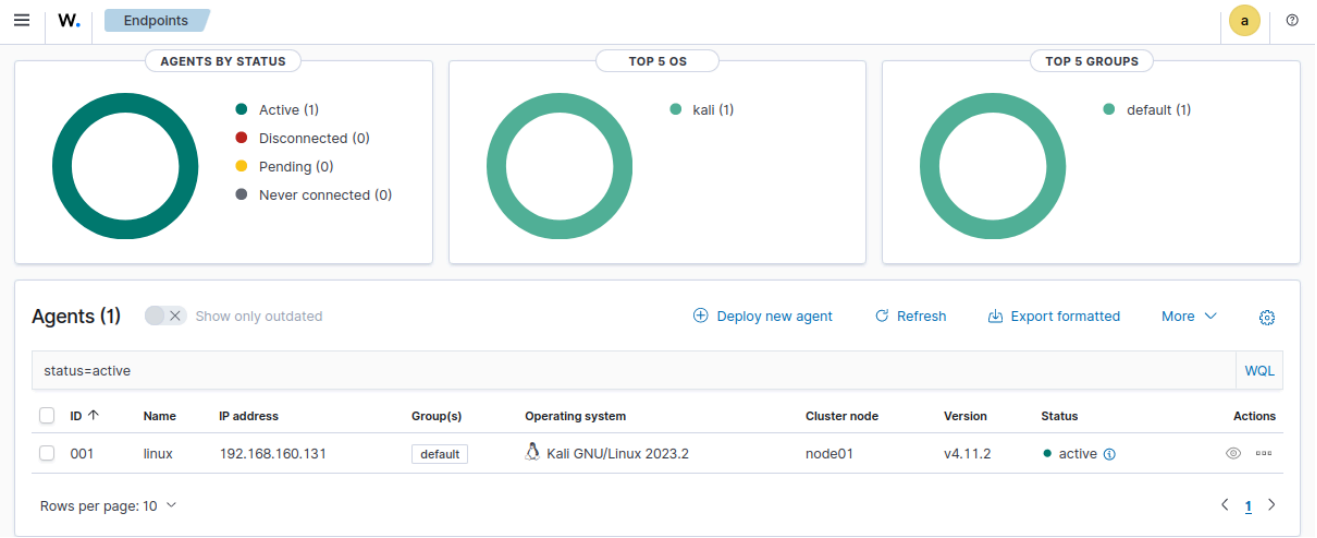



```
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.11.2-1_amd64.deb && sudo WAZUH_MANAGER='192.168.160.129' WAZUH_AGENT_NAME='linux' dpkg -i ./wazuh-agent_4.11.2-1_amd64.deb
```

```
┌──(kali㉿kali)-[~]
└─$ wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.11.2-1_amd64.deb && sudo WAZUH_MANAGER='192.168.160.129' WAZUH_AGENT_NAME='linux' dpkg -i ./wazuh-agent_4.11.2-1_amd64.deb
--2025-04-14 16:49:54--  https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.11.2-1_amd64.deb
Resolving packages.wazuh.com (packages.wazuh.com)... 52.84.45.85, 52.84.45.38, 52.84.45.66, ...
Connecting to packages.wazuh.com (packages.wazuh.com)|52.84.45.85|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11075686 (11M) [application/vnd.debian.binary-package]
Saving to: 'wazuh-agent_4.11.2-1_amd64.deb'

wazuh-agent_4.11.2-1_amd64.deb  100%[═══════════════════════════════════════════]  10.56M   902KB/s    in 17s

2025-04-14 16:50:13 (632 KB/s) - 'wazuh-agent_4.11.2-1_amd64.deb' saved [11075686/11075686]

Selecting previously unselected package wazuh-agent.
(Reading database ... 397970 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.11.2-1_amd64.deb ...
Unpacking wazuh-agent (4.11.2-1) ...
Setting up wazuh-agent (4.11.2-1) ...

┌──(kali㉿kali)-[~]
└─$
```

```
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
```

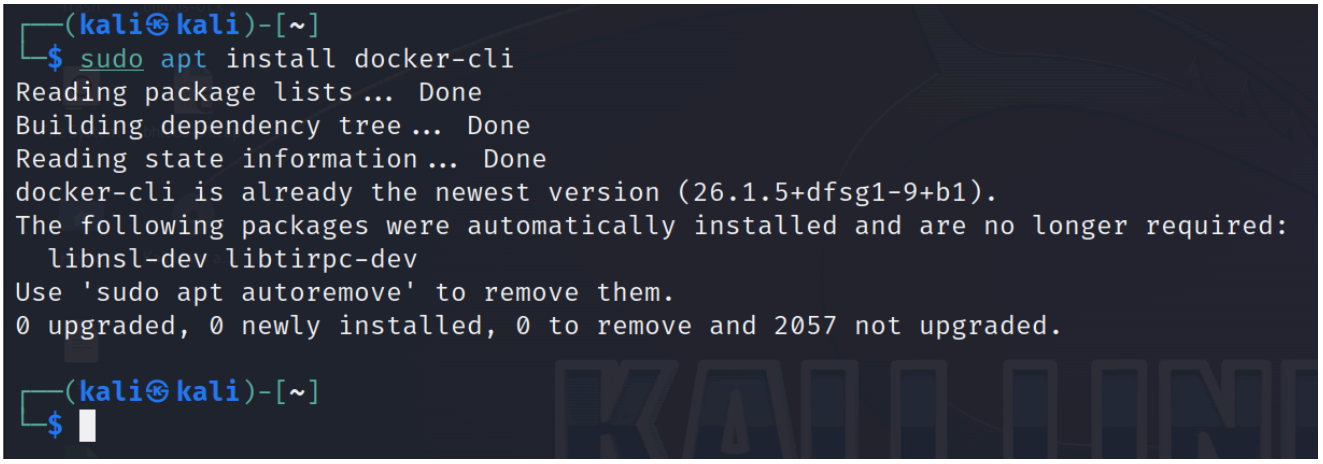
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Created symlink '/etc/systemd/system/multi-user.target.wants/wazuh-agent.service' → '/usr/lib/systemd/system/wazuh-agent.service'.
```



The Wazuh dashboard confirms successful agent deployment, showing our Linux agent (192.168.160.131) as active and properly reporting under the default group.

Exercise 2: Monitoring Docker events

Configure the monitored endpoint



The output confirms Docker CLI is already installed on our Kali system. Now we can proceed with configuring Wazuh to monitor Docker events

To activate Docker event collection, we added this configuration to `/var/ossec/etc/ossec.conf`:

```
<wodle name="docker-listener">
  <interval>10m</interval>
  <attempts>5</attempts>
  <run_on_start>yes</run_on_start>
  <disabled>no</disabled>
</wodle>
```

```
<ossec_config>
  <wodle name="docker-listener">
    <interval>10m</interval>
    <attempts>5</attempts>
    <run_on_start>yes</run_on_start>
    <disabled>no</disabled>
  </wodle>
</ossec_config>
```

```
┌──(root㉿kali)-[/var/ossec/etc]
└─# sudo systemctl restart wazuh-agent
```

Test the configuration

```
sudo docker pull nginx
```

```
┌──(kali㉿kali)-[/]
└─$ sudo docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
8a628cdd7ccc: Pull complete
75b642592991: Pull complete
553c8756fd66: Pull complete
10fe6d2248e3: Pull complete
3b6e18ae4ce6: Pull complete
3dce86e3b082: Pull complete
e81a6b82cf64: Pull complete
Digest: sha256:09369da6b10306312cd908661320086bf87fbae1b6b0c49a1f50ba531fef2eab
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

```
sudo docker run -d -P --name nginx_container nginx
```

```
┌──(kali㉿kali)-[/]
└─$ sudo docker run -d -P --name nginx_container nginx
ad79dd90379329fe44fda8a955c97ed0069e310907426f068814682859990f7f
```

```
sudo docker exec -it nginx_container cat /etc/passwd
```

```
┌──(kali㉿kali)-[/]
└─$ sudo docker exec -it nginx_container cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
nginx:x:101:101:nginx user:/nonexistent:/bin/false
```

```
sudo docker exec -it nginx_container /bin/bash
```

```
┌──(kali㉿kali)-[/]
└─$ sudo docker exec -it nginx_container /bin/bash
root@ad79dd903793:/# ls
bin   dev                  docker-entrypoint.sh  home  lib64  mnt  proc  run   srv  tmp  var
boot  docker-entrypoint.d  etc                   lib   media  opt  root  sbin  sys  usr
root@ad79dd903793:/# pwd
/
root@ad79dd903793:/# whoami
root
root@ad79dd903793:/# exit
exit
```

```
sudo docker stop nginx_container
```

```
┌──(kali㉿kali)-[/]
└─$ sudo docker stop nginx_container
nginx_container
```

```
sudo docker rm nginx_container
```

```
┌──(kali㉿kali)-[/]
└─$ sudo docker rm nginx_container
nginx_container
```

After running all the tests we can check that we correctly configured everything by checking the **linux-agent** logs in the web interface

13 hits

Apr 13, 2025 @ 16:06:51.281 - Apr 14, 2025 @ 16:06:51.281

Export Formatted | 638 available fields | Columns | Density | **1 fields sorted** | Full screen

timestamp	agent.name	data.docker.from	data.docker.Type	data.docker.Action	rule.description	rule.level
Apr 14, 2025 @ 16:03:40.305	linux	nginx	container	destroy	Docker: Container nginx_container destroyed	5
Apr 14, 2025 @ 16:03:34.717	linux	nginx	container	die	Docker: Container nginx_container received the action: die	7
Apr 14, 2025 @ 16:03:34.697	linux	nginx	container	stop	Docker: Container nginx_container stopped	3
Apr 14, 2025 @ 16:03:34.631	linux	-	network	disconnect	Docker: Network bridge disconnected	4