

In this task, we are going to see how to installing Docker and Elastic on an Ubuntu machine, followed by verifying network activity through ping and nmap scans.

The screenshot shows a blue header with the Dockerdocs logo, navigation links for 'Get started', 'Guides', 'Manuals' (which is underlined), and 'Reference', and an 'Ask AI' button. Below the header, there's a sidebar titled 'OPEN SOURCE' with a dropdown menu for 'Docker Engine' containing 'Install' (which is underlined) and sub-options for 'Ubuntu' (selected and highlighted in grey), 'Debian', and 'RHEL'. The main content area has a breadcrumb trail: 'Home / Manuals / Docker Engine / Install / Ubuntu'. The title 'Install Docker Engine on Ubuntu' is underlined. A note below the title says: 'To get started with Docker Engine on Ubuntu, make sure you [meet the prerequisites](#), and then follow the [installation steps](#).'. There's also a 'Page options' button.

We received a document that teaches how to install Docker in ubuntu, let's just follow the document.

```
brandon@brandon-HP-EliteDesk:~/Docker$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
ea52d2000f90: Download complete
Digest: sha256:f7931603f70e13dbd844253370742c4fc4202d290c80442b2e68706d8f33ce26
Status: Downloaded newer image for hello-world:latest

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

When you finish installing Docker, you'll see this image.

In the next step, we have to install the Elastic stack, It is a research, logging, and observability platform widely used in security, DevOps, and data analytics.



Supercharge your skills at ElasticON – join us at an event near you. Register now!

The open source platform that powers search, observability, security, and more ...

Build with Elasticsearch

That's the interface of the site that we need.

The screenshot shows the Elasticsearch homepage with a dark blue background. In the center, the text "Elasticsearch — the most widely deployed vector database" is displayed in white. Below this, there are two main calls to action: one on the left for local deployment and one on the right for production deployment. A red box highlights the terminal command for local deployment. At the bottom, there are links to documentation and a download option.

Elasticsearch — the most widely deployed vector database

Copy to try locally in two minutes

```
curl -fsSL https://elastic.co/start-local | sh
```

Deploy for production

Start free cloud trial

OR

Read docs →

Or, [download on-prem](#)

We just have to scroll down to the bottom of the website, copy and paste this code into our terminal.

```

[{"id": 1, "x": 100, "y": 100, "w": 100, "h": 100}, {"id": 2, "x": 200, "y": 100, "w": 100, "h": 100}, {"id": 3, "x": 300, "y": 100, "w": 100, "h": 100}, {"id": 4, "x": 100, "y": 200, "w": 100, "h": 100}, {"id": 5, "x": 200, "y": 200, "w": 100, "h": 100}, {"id": 6, "x": 300, "y": 200, "w": 100, "h": 100}, {"id": 7, "x": 100, "y": 300, "w": 100, "h": 100}, {"id": 8, "x": 200, "y": 300, "w": 100, "h": 100}, {"id": 9, "x": 300, "y": 300, "w": 100, "h": 100}, {"id": 10, "x": 100, "y": 400, "w": 100, "h": 100}, {"id": 11, "x": 200, "y": 400, "w": 100, "h": 100}, {"id": 12, "x": 300, "y": 400, "w": 100, "h": 100}, {"id": 13, "x": 100, "y": 500, "w": 100, "h": 100}, {"id": 14, "x": 200, "y": 500, "w": 100, "h": 100}, {"id": 15, "x": 300, "y": 500, "w": 100, "h": 100}], [{"x": 100, "y": 100, "x2": 200, "y2": 100}, {"x": 200, "y": 100, "x2": 300, "y2": 100}, {"x": 100, "y": 200, "x2": 200, "y2": 200}, {"x": 200, "y": 200, "x2": 300, "y2": 200}, {"x": 100, "y": 300, "x2": 200, "y2": 300}, {"x": 200, "y": 300, "x2": 300, "y2": 300}, {"x": 100, "y": 400, "x2": 200, "y2": 400}, {"x": 200, "y": 400, "x2": 300, "y2": 400}, {"x": 100, "y": 500, "x2": 200, "y2": 500}, {"x": 200, "y": 500, "x2": 300, "y2": 500}, {"x": 100, "y": 100, "x2": 100, "y2": 200}, {"x": 200, "y": 100, "x2": 200, "y2": 200}, {"x": 300, "y": 100, "x2": 300, "y2": 200}, {"x": 100, "y": 200, "x2": 100, "y2": 300}, {"x": 200, "y": 200, "x2": 200, "y2": 300}, {"x": 300, "y": 200, "x2": 300, "y2": 300}, {"x": 100, "y": 300, "x2": 100, "y2": 400}, {"x": 200, "y": 300, "x2": 200, "y2": 400}, {"x": 300, "y": 300, "x2": 300, "y2": 400}, {"x": 100, "y": 400, "x2": 100, "y2": 500}, {"x": 200, "y": 400, "x2": 200, "y2": 500}, {"x": 300, "y": 400, "x2": 300, "y2": 500}, {"x": 100, "y": 100, "x2": 100, "y2": 100}, {"x": 200, "y": 100, "x2": 200, "y2": 100}, {"x": 300, "y": 100, "x2": 300, "y2": 100}, {"x": 100, "y": 200, "x2": 100, "y2": 200}, {"x": 200, "y": 200, "x2": 200, "y2": 200}, {"x": 300, "y": 200, "x2": 300, "y2": 200}, {"x": 100, "y": 300, "x2": 100, "y2": 300}, {"x": 200, "y": 300, "x2": 200, "y2": 300}, {"x": 300, "y": 300, "x2": 300, "y2": 300}, {"x": 100, "y": 400, "x2": 100, "y2": 400}, {"x": 200, "y": 400, "x2": 200, "y2": 400}, {"x": 300, "y": 400, "x2": 300, "y2": 400}, {"x": 100, "y": 500, "x2": 100, "y2": 500}, {"x": 200, "y": 500, "x2": 200, "y2": 500}, {"x": 300, "y": 500, "x2": 300, "y2": 500}], [{"text": "Run Elasticsearch and Kibana for local testing"}], [{"text": "\u26a0 Do not use this script in a production environment"}, {"text": "\u26a0 Setting up Elasticsearch and Kibana v9.2.2..."}, {"text": "- Generated random passwords"}, {"text": "- Created the elastic-start-local folder containing the files:"}, {"text": "- .env, with settings"}, {"text": "- docker-compose.yml, for Docker services"}, {"text": "- start/stop/uninstall commands"}, {"text": "- Running docker compose up --wait"}]

```

When you run the command, you will see this image.

The screenshot shows the Fleet interface with the following details:

- Fleet** heading
- Centralized management for Elastic Agents.
- Agents tab selected.
- Filter bar: Filter your data using KQL syntax, Status (3), Tags (1), Agent policy (1), Upgrade available.
- Agent activity section: Shows 1 agent.
- Table view:

Status	Host	Agent policy	CPU	Memory	Last activity	Version	Actions
Healthy	1e94558c72c4	Elastic Cloud agent policy rev. 4	N/A	N/A	32 seconds ago	9.2.2	...
- Page controls: Rows per page: 20, navigation arrows.

Now, let's add an agent in our pc, we have to go to feet in the bweb-site and click in add feet server.

X

Add a Fleet Server

A Fleet Server is required before you can enroll agents with Fleet. Follow the instructions below to set up a Fleet Server. For more information, see the [Fleet and Elastic Agent Guide](#)

Quick Start

Advanced

1 Get started with Fleet Server

First, set the public IP or host name and port that agents will use to reach Fleet Server. It uses port **8220** by default ②. We'll then generate a policy for you automatically.

Name _____

fleet server

URL

<https://localhost:8220>

 Add another URL

Make this Fleet server the default one.

Continue

We have to add a new server for our agent, just give a name and put this url.

Showing 2 agents		Clear filters		Healthy	Unhealthy	Orphaned	Updating	Offline	Inactive	Unenrolled	Uninstalled
Status	Host	Agent policy	CPU	Memory	Last activity	Version	Actions				
Healthy	brandon-HP-EliteDesk rev. 1	Fleet Server Policy	3.19 %	251 MB	17 seconds ago	9.2.2	...				
Healthy	1e94558c72c4 rev. 4	Elastic Cloud agent policy	N/A	N/A	14 seconds ago	9.2.2	...				

Here we can see our agent.

The screenshot shows the Elasticsearch interface with the 'Deployment' tab selected. On the left, there's a sidebar with icons for Elasticsearch, Discover, Dashboards, Playground, Machine Learning, and various logs (alerts, file, library, network, process, registry, security). The main content area is titled 'Elastic Defend' under the 'Elastic Agent' section. It has tabs for 'Overview' (which is selected), 'Settings', and 'Advanced'. The 'Overview' tab displays the 'Elastic Defend Detection and Response Dashboard' which allows users to instantly investigate, triage, and respond to threats. Below this, the 'Elastic Defend Integration' section provides details about its capabilities: prevention, detection, and response across Windows, macOS, and Linux operating systems. It lists four key features: preventing complex attacks, alerting in high fidelity, detecting threats in high fidelity, and triaging and responding rapidly. The page also mentions that it uses cases across EPP, EDR, SIEM, and Security Analytics.

We have to look for Elastic-defend and do the download.

```
brandon@brandon-HP-EliteDesk:~/elastic-agent-9.2.2-linux-arm64/elastic-agent-9.2.2-linux-x86_64$ nmap 192.168.1.12
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-12-09 12:26 WET
Nmap scan report for brandon-HP-EliteDesk.home (192.168.1.12)
Host is up (0.00029s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
3389/tcp  open  ms-wbt-server
8080/tcp  open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

Let's do a scan with nmap to see if it will appear in Elastic application.

The screenshot shows a list of log entries from the Elasticsearch 'Logs' section. The first entry is highlighted with a red box around the timestamp 'Dec 9, 2025 @ 12:26:42.903'. The log message contains several fields: process.name (nmap), @timestamp (Dec 9, 2025 @ 12:26:42.903), agent.id (52e1c5ec-42c8-4335-9ca9-670cdxfc136e), agent.type (endpoint), agent.version (9.2.2), data_stream.dataset (endpoint.events.network), data_stream.namespace (default), data_stream.type (logs), destination.address (192.168.1.12), destination.bytes (0), and destination.ip (192.168.1.12). There are five more similar log entries below it, each with a different timestamp but identical structure.

Perfect, we can check the scan, and others details, like date, hour and destination.

```
rm64/elastic-agent-9.2.2-linux-x86_64$ nmap 192.168.1.12 -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-12-09 12:45 WET
Nmap scan report for brandon-HP-EliteDesk.home (192.168.1.12)
Host is up (0.00021s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 9.6p1 Ubuntu 3ubuntu13.14 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 63:0b:24:fd:be:f9:41:0a:b3:66:71:34:1c:46:0e:df (ECDSA)
|_  256 41:98:a0:a6:fd:61:19:92:29:4f:76:0b:c5:47:da:66 (ED25519)
3389/tcp  open  ms-wbt-server?
8080/tcp  open  http-proxy
| http-open-proxy: Proxy might be redirecting requests
```

Now, we are going to use a filter -A to see what happens in the Elastic logs.

The screenshot shows the Elasticsearch interface with a search results page. The top navigation bar includes 'Documents (1,002)', 'Patterns', and 'Field statistics'. On the right, there are icons for sorting, search, and other functions. The main area displays a list of log entries. One entry is expanded to show its full JSON structure:

```

{
  "process.default-2025.12.09-000001": {
    "_id": "AZsDjxx4AmKWFt7r0Qp",
    "_version": 1,
    "_score": null,
    "fields": {
      "process.command_line.caseless": [
        "nmap 192.168.1.12 -A"
      ],
      "event.category": [
        "process"
      ]
    }
  }
}
```

Below the expanded log, there is a 'Copy value' button and a note: 'agent.type endpoint agent.version 9.2.2 data_stream.dataset endpoint.events.network data_stream.namespace defa'. The bottom of the screen shows a pagination bar with pages 1 through 5.

If we expand the log, we can check the command that was used and his filter.

