

# Install Runtime Sensor for Linux (native)



Use this guide to install Sensors on Linux hosts, natively or via Docker.

## Before you begin

The prerequisites are:

- [Supported Linux distribution and kernel versions](#).
- Make sure you have the [required permissions](#).
- Grant access to the [required URLs](#).
- (Docker only) [Get the Runtime Sensor image pull key from Wiz](#), required for the `PULL_USERNAME` and `PULL_PASSWORD` parameters in the script below.

## Installation steps

[Step 1](#): Create a Service Account for the Runtime Sensor in Wiz

[Step 2](#): Run the installation script


[Step 3](#): (Optional) Validate the Runtime Sensor is running

[Step 4](#): See your Sensors in Wiz

## Create a Service Account for the Runtime Sensor in Wiz

Create a service account the Runtime Sensor can use to communicate with the Wiz backend.

1. In Wiz, navigate to [Settings > Access Management > Service Accounts](#), then click Add Service Account.
2. Enter a meaningful name for the account.
3. From Type, select Sensor or Complete Kubernetes Integration ([learn about the different types of Service Accounts](#)). Click Add Service Account.
4. A dialog window opens with your new OAuth credentials. Copy the Client ID and Client secret, you will need them later on.

 The Client ID and Client Secret are available only once! Be sure to save them in a secure place.

5. Click Finish

## Run the installation script

Follow the steps in this procedure to review and configure your environment parameters, invoke the script, and verify execution.

1. Review the high-level workflow of the Runtime Sensor installation script you are using:


### ▼ Native Linux

1. Verifies the current machine is supported by detecting the kernel version and Linux distribution.
2. Writes a configuration file with your environment parameters, which will be later sent to the Runtime Sensor.
3. Adds a new package repository to the native package manager (APT/YUM) with an appropriate GPG key to verify integrity.
4. Installs the `wiz-sensor` package using the native package manager.
5. Starts the `wiz-sensor` service that was installed as part of the package.

### ▼ Docker

1. Verifies the current machine is supported by detecting the kernel version and Linux distribution.
2. Confirms that `docker` and `systemd` are installed.
3. Authenticates with the Wiz container registry.
4. Installs and starts the `wiz-sensor` `systemd` service, designed to retrieve and execute the Sensor using Docker.

2. [See the list of all supported environment variables](#). Review them and plan your deployment accordingly.

 The `WIZ_API_CLIENT_ID` and `WIZ_API_CLIENT_SECRET` parameters are mandatory and must be included in your installation command. All other parameters are optional.

3. Invoke the installation script. Our example includes the most common parameters:

Select install method ☐ Native Linux ☐ Docker



- The script must be run as root (if required, use `sudo` or a similar utility). Remember that env-variables are not automatically inherited when switching users.
- The script requires the `curl` utility to be installed on the machine.

## (Optional) Validate the Sensor is installed and running

### Validate the script finished running successfully

Below is the output of a successful script execution:

#### ▼ RedHat-based distributions

Shell

```
Existing host keys found in
/home/*****/.ssh/google_compute_known_hosts
^@^@ % Total      % Received % Xferd  Average Speed   Time    Time
Time  Current
                                Dload  Upload  Total  Spent  Left
Speed
100 17124  100 17124    0     0  40654      0 --:--:-- --:--:-- --:--:--
40771
```

```

  _ _ _ _ _
 \ \      / (_)___ / ___| | _ _ _ _ _
  \ \ / \ / / | | / \___ \ / _ \ ' _ \ / _ \
   \ V V / | | / / ___) | | | | \_ \ \ ( ) | |
    \/\ / | | /___| |___ \___/ \___/ \___/ | |
```

```
Detected OS is RedHat
Installing YUM sources for Wiz
Cache was expired
5 files removed
Wiz, Inc.                                62 kB/s | 68 kB
00:01
Dependencies resolved.
```

```
=====
Package          Architecture Version           Repository
Size
```



```

Hit:3 https://deb.debian.org/debian bullseye InRelease
Hit:4 https://deb.debian.org/debian-security bullseye-security InRelease
Hit:5 https://deb.debian.org/debian bullseye-updates InRelease
Hit:6 https://deb.debian.org/debian bullseye-backports InRelease
Get:7 https://dpkg.wiz.io/projects/sensor-repos sensor-apt-dev InRelease
[1301 B]
Get:8 https://dpkg.wiz.io/projects/sensor-repos sensor-apt-dev/main
arm64 Packages [93.2 kB]
Fetched 94.5 kB in 1s (80.9 kB/s)
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
apt-transport-https is already the newest version (2.2.4).
gnupg is already the newest version (2.2.27-2+deb11u2).
0 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
Installing APT sources for Wiz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current                                  Dload  Upload   Total   Spent    Left
Speed
100 1021  100 1021    0     0  9116      0 --:--:-- --:--:-- --:--:--
9116
gpg: key C0BA5CE6DC6315A3: "Artifact Registry Repository Signer
<artifact-registry-repository-signer@google.com>" not changed
add: Total number processed: 1

```

## **Docker**

Shell

```

  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current                                  Dload  Upload   Total   Spent    Left
Speed
100 17124  100 17124    0     0 45792      0 --:--:-- --:--:-- --:--:--
45786

  _ _ _ _ _
 \ \   / /  / ( _ ) _ _ / _ _ | _ _ _ _ _
  \ \ / /  / / | _ / \ _ _ \ / _ \ ' _ \ | _ _ |
   \ V V /  | / / _ _ ) | _ / | | \ _ _ \ ( _ ) | |
    \ _ \ /  |/_ _ | | _ _ / \ _ _ | | | _ _ / \ _ _ | |

```

```

Detected OS is RedHat
Created symlink from /etc/systemd/system/multi-user.target.wants/wiz-
sensor.service to /etc/systemd/system/wiz-sensor.service.
Wiz sensor was successfully installed!

```

## Validate the Runtime Sensor is running

After the Sensor is installed (installation usually takes up to one minute), it will start running. You can verify this using `ps` utility:

### ▼ Native Linux

Shell

```
vagrant@vagrant:~$ ps aux | grep wiz
2526 ?        Ss          0:00 /opt/wiz/sensor/sensor_init run
2563 ?        S           0:00 /opt/wiz/sensor/sensor_init sensor_env
/opt/wiz/sensor/host-store/engine.tar.xz /usr/src/app/wiz-sensor daemon --
json
2586 ?        Sl          0:00 /usr/src/app/wiz-sensor daemon --json
2607 ?        Sl          0:16 /usr/src/app/wiz-sensor daemon --json --run-
engine
2704 pts/0    S+          0:00 grep --color=auto wiz
```

### ▼ Docker

Shell

```
user@ubuntu-vm /tmp $ ps aux | grep wiz
48892 ?        Ssl         0:00 docker run --name wiz-sensor --restart
unless-stopped --mount
type=bind,source=/sys/kernel/debug,target=/sys/kernel/debug,readonly --
mount type=tmpfs,destination=/tmp,tmpfs-size=100m --env-file
/opt/wiz/sensor/host-store/sensor_config.env -v /opt/wiz/sensor/host-
store:/wiz-host-cache -u 2202:2202 --cgroupns host --pid host --ipc host -
-network host --read-only --security-opt apparmor=unconfined --security-
opt seccomp=unconfined --security-opt label:disable --cap-add=SYS_ADMIN --
cap-add=SYS_CHROOT --cap-add=SYS_RESOURCE --cap-add=SYS_RAWIO --cap-
add=DAC_OVERRIDE --cap-add=DAC_READ_SEARCH --cap-add=NET_ADMIN --cap-
add=NET_RAW --cap-add=IPC_LOCK --cap-add=FOWNER --cap-add=SYS_PTRACE --
cap-add=KILL --cpus=0.3 --memory=300M wizio.azurecr.io/sensor:v1
48927 ?        Ssl         0:00 ./wiz-sensor daemon --json
48970 ?        Sl          0:01 /usr/src/app/wiz-sensor daemon --json --run-
engine
49060 pts/0    S+          0:00 grep --colour=auto wiz
```

## See your Sensors in Wiz

From the Wiz portal, verify successful Sensor installation and operation:

1. View all installed Sensors and their statuses ([direct link](#))
2. Address any System Health Issues ([direct link](#))
3. Explore all Sensor Threat Detection Rules and Issues ([direct link](#))

 Updated 2 months ago

---

[← Runtime Sensor for Linux](#)

Install Runtime Sensor for Linux  
(ECS on EC2) [→](#)

Did this page help you?  **Yes**  **No**