Wazuh indexer

The Wazuh indexer is a real-time, full-text search and analytics engine for security data. Log data ingested into the Wazuh server is analyzed and forwarded to the indexer for indexing and storage. These events are then queried on the Wazuh dashboard.

The Wazuh indexer stores data as JSON documents. Each document associates a set of keys, field names, or attributes with their corresponding values, which can be characters, numbers, booleans, dates, arrays of values, geolocations, or other kinds of data.

The Wazuh indexer can be configured as a single-node or multi-node cluster, providing scalability and high availability. It distributes documents across different containers, known as shards. In turn, it distributes these shards across cluster nodes. By distributing the documents across multiple shards and distributing those shards across multiple nodes, the Wazuh indexer ensures redundancy. Redundancy ensures the Wazuh indexer's availability in the event of a failure and boosts query capacity across cluster nodes.

**Contents**

* [Wazuh indexer indices](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html)
* [Re-indexing](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/re-indexing.html)
* [Wazuh indexer tuning](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-tuning.html)
* [Migrating Wazuh indices](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/migrating-wazuh-indices.html)
* [Wazuh indexer configuration on hardened endpoints](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-on-hardened-endpoints.html)

# Wazuh indexer indices

An index is a collection of documents that relate to each other. The Wazuh indexer uses indices to store and organize security data for fast retrieval. Wazuh uses the following index patterns to store this data:

* [wazuh‑alerts-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-alerts-indices): This is the index pattern for alerts generated by the Wazuh server.
* [wazuh‑archives-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-archives-indices): This is the index pattern for all events sent to the Wazuh server.
* [wazuh‑monitoring-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-monitoring-indices): This is the index pattern for the status of the Wazuh agents.
* [wazuh‑statistics-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-statistics-indices): This is the index pattern that shows the performance metrics of the Wazuh server. It contains information that shows how many events are received, processed and dropped by the Wazuh server.
* [wazuh-states-vulnerabilities-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-states-vulnerabilities-indices): This is the index pattern for information about vulnerabilities detected in the endpoints being monitored.
* [wazuh-states-inventory-hardware-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-states-inventory-hardware-indices): This is the index pattern for basic information about hardware components on a monitored endpoint.
* [wazuh-states-inventory-hotfixes-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-states-inventory-hotfixes-indices): This is the index pattern for updates installed on a Windows endpoint. The Wazuh vulnerability detection module uses this to discover what vulnerabilities have been patched on a Windows endpoint.
* [wazuh-states-inventory-interfaces-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-states-inventory-interfaces-indices): This is the index pattern for up and down status information, and packet transfer information about network interfaces on a monitored endpoint.
* [wazuh-states-inventory-networks-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-states-inventory-netwworks-indices): This is the index pattern for IPv4 and IPv6 addresses associated with each network interface on a monitored endpoint.
* [wazuh-states-inventory-packages-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-states-inventory-packages-indices): This is the index pattern for currently installed software packages on an endpoint.
* [wazuh-states-inventory-ports-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-states-inventory-ports-indices): This is the index pattern for open network ports on a monitored endpoint.
* [wazuh-states-inventory-processes-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-states-inventory-processes-indices): This is the index pattern for system processes running on a monitored endpoint.
* [wazuh-states-inventory-protocols-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-states-inventory-protocols-indices): This is the index pattern for network routing configuration details and protocols for each network interface on a monitored endpoint.
* [wazuh-states-inventory-system-\*](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-indices.html#wazuh-states-inventory-system-indices): This is the index pattern for information on the operating system, hostname, and architecture on a monitored endpoint.

To further customize the index pattern for alerts, you can create a custom index pattern.

## Creating custom index pattern

This section describes how to create a custom index pattern, for example, my-custom-alerts-\*, alongside the default pattern, wazuh-alerts-\*. Switch to the root user and perform the steps below.

1. Stop the Filebeat service:
2. systemctl stop filebeat
3. Download the Wazuh template and save it into a file (for example, template.json):
4. curl -so template.json https://raw.githubusercontent.com/wazuh/wazuh/v4.13.1/extensions/elasticsearch/7.x/wazuh-template.json
5. Open the template file and locate this line at the beginning of the file:
6. "index\_patterns": [
7. "wazuh-alerts-4.x-\*",
8. "wazuh-archives-4.x-\*"
9. ],

Add your custom pattern to look like this:

"index\_patterns": [

"wazuh-alerts-4.x-\*",

"wazuh-archives-4.x-\*",

"my-custom-alerts-\*"

],

The asterisk character (\*) on the index patterns is important because Filebeat will create indices using a name that follows this pattern, which is necessary to apply the proper format to visualize the alerts on the Wazuh dashboard.

1. Save the modifications and insert the new template into the Wazuh indexer. This will replace the existing template:
2. curl -XPUT -k -u <INDEXER\_USERNAME>:<INDEXER\_PASSWORD> 'https://<INDEXER\_IP\_ADDRESS>:9200/\_template/wazuh' -H 'Content-Type: application/json' -d @template.json

Replace:

* + <INDEXER\_IP\_ADDRESS> with the IP address of your Wazuh indexer
  + <INDEXER\_USERNAME> and <INDEXER\_PASSWORD> with the Wazuh indexer username and password. You can obtain the Wazuh indexer credentials for fresh deployments using the command:

**Note**

If using the Wazuh OVA, use the default credentials admin:admin or refer to the [password management](https://documentation.wazuh.com/current/user-manual/user-administration/password-management.html) section.

tar -axf wazuh-install-files.tar wazuh-install-files/wazuh-passwords.txt -O | grep -P "\'admin\'" -A 1

{"acknowledged":true}

**Note**

{"acknowledged":true} indicates that the template was inserted correctly.

**Warning**

Perform step 5 only if you want to replace the default alert index pattern wazuh-alerts-\* and/or the default archive index pattern wazuh‑archives-\* with my-custom-alerts-\*.

1. Open the Wazuh alerts configuration file /usr/share/filebeat/module/wazuh/alerts/manifest.yml and optionally the archives file /usr/share/filebeat/module/wazuh/archives/manifest.yml and replace the index name.

For example, from:

- name: index\_prefix

default: wazuh-alerts-

To this:

- name: index\_prefix

default: my-custom-alerts-

**Note**

The index name must not contain the characters #, \, /, \*, ?, ", <, >, |, ,, and must not start with \_, -, or +. Also, all the letters must be lowercase.

1. (Optional) If you want to use the new index pattern by default, open the /usr/share/wazuh-dashboard/data/wazuh/config/wazuh.yml file and add the below configuration:
2. pattern: my-custom-alerts-\*

This will make the Wazuh server automatically create and/or select the new index pattern.

1. Restart Filebeat and the Wazuh server components:
2. systemctl restart filebeat
3. systemctl restart wazuh-manager
4. systemctl restart wazuh-indexer
5. systemctl restart wazuh-dashboard

**Warning**

If you already have indices created with the previous name, they won't be changed. You can still change to the previous index pattern to see them, or you can perform [reindexing](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/re-indexing.html) to rename the existing indices.

## Checking indices information

You can check for information about Wazuh indices in two ways.

* Using the web user interface.
* Making a request to the Wazuh indexer API.

### Using the web user interface

1. In the Wazuh dashboard upper left menu **☰**, go to **Indexer management** > **Index Management**.
2. Click on **Indices**.

If the pattern is not present in the Wazuh dashboard, create a new one using the index pattern used in the template my-custom-alerts-\*, and make sure to use timestamp as the **Time Filter** field name.

### Using the Wazuh indexer API

You can query the indices information using the Wazuh indexer API from the Wazuh dashboard or the Wazuh server.

#### Wazuh dashboard

1. Navigate to **☰** > **Indexer management** > **Dev Tools**:
2. GET /\_cat/indices/wazuh-\*?v

#### Command line interface

1. Obtain the Wazuh indexer username and password for fresh deployments using the below command:
2. tar -axf wazuh-install-files.tar wazuh-install-files/wazuh-passwords.txt -O | grep -P "\'admin\'" -A 1

**Note**

If using the Wazuh OVA, use the default credentials admin:admin or refer to the [password management](https://documentation.wazuh.com/current/user-manual/user-administration/password-management.html) section.

1. Run the following command to query your index status. Replace <INDEXER\_USERNAME> and <INDEXER\_PASSWORD> with the username and password obtained. Replace <INDEXER\_IP\_ADDRESS> with your Wazuh indexer IP address or FQDN. You can replace wazuh-\* with a more specific pattern for your query, such as wazuh-alerts-\*.
2. curl -k -u <INDEXER\_USERNAME>:<INDEXER\_PASSWORD> https://<INDEXER\_IP\_ADDRESS>:9200/\_cat/indices/wazuh-\*?v
3. health status index uuid pri rep docs.count docs.deleted store.size pri.store.size
4. green open wazuh-statistics-2023.30w xtHZtGqBR0WNJWbs5sjrnQ 1 0 2394 0 1.2mb 1.2mb
5. green open wazuh-alerts-4.x-2023.07.28 VbBfAasJTsiqw3lwRhY5sg 3 0 513 0 1.9mb 1.9mb
6. green open wazuh-alerts-4.x-2023.07.27 7s2x8INqRVmtz5uqMDuA7Q 3 0 515 0 2mb 2mb
7. green open wazuh-alerts-4.x-2023.07.05 0h4cyLJoQYiMvMnqyLDnag 3 0 49 0 370.4kb 370.4kb
8. green open wazuh-alerts-4.x-2023.07.07 kp\_N4c7RRuOE91KkuqPuAw 3 0 98 0 397.7kb 397.7kb
9. green open wazuh-alerts-4.x-2023.07.29 rbAC4befS7epxOjiSzFRQQ 3 0 1717 0 3.9mb 3.9mb
10. green open wazuh-monitoring-2023.31w 1WwxsGQHRfG1\_DOIZD-Lag 1 0 954 0 771.9kb 771.9kb
11. green open wazuh-alerts-4.x-2023.07.20 SQbaQC24SgO9eWO\_AsBI\_w 3 0 1181 0 2.8mb 2.8mb
12. green open wazuh-statistics-2023.28w jO52bS6eRamtB2YNmfGzIA 1 0 676 0 501.1kb 501.1kb

## The wazuh‑alerts-\* indices

The Wazuh server analyzes events received from monitored endpoints and generates alerts when the events match a detection rule. These alerts are saved using the wazuh-alerts-\* indices.

The Wazuh server logs the alert data into the /var/ossec/logs/alerts/alerts.json and /var/ossec/logs/alerts/alerts.log files by default. Once saved in the /var/ossec/logs/alerts/alerts.json file, it forwards the JSON alert document to the Wazuh indexer API for indexing. The indexed files are stored in the /var/lib/wazuh-indexer/nodes/0/indices directory of the Wazuh indexer.

When forwarding alerts to the Wazuh indexer, the Wazuh server formats the current date into an index name. For example, the Wazuh server will define the index names wazuh-alerts-4.x-2023.03.17 and wazuh-alerts-4.x-2023.03.18 for March 17th and 18th alerts, respectively. The Wazuh indexer then creates alert indices using the defined wazuh‑alerts-\* index names.

You can modify the default index name in the /usr/share/filebeat/module/wazuh/alerts/ingest/pipeline.json file of the Wazuh server. To do this, navigate to the date\_index\_name field and date\_rounding key to change the default index name formatting in the /usr/share/filebeat/module/wazuh/alerts/ingest/pipeline.json file:

{

"description": "Wazuh alerts pipeline",

"processors": [

{ "json" : { "field" : "message", "add\_to\_root": true } },

{

"geoip": {

"field": "data.srcip",

"target\_field": "GeoLocation",

"properties": ["city\_name", "country\_name", "region\_name", "location"],

"ignore\_missing": true,

"ignore\_failure": true

}

},

{

"geoip": {

"field": "data.win.eventdata.ipAddress",

"target\_field": "GeoLocation",

"properties": ["city\_name", "country\_name", "region\_name", "location"],

"ignore\_missing": true,

"ignore\_failure": true

}

},

{

"geoip": {

"field": "data.aws.sourceIPAddress",

"target\_field": "GeoLocation",

"properties": ["city\_name", "country\_name", "region\_name", "location"],

"ignore\_missing": true,

"ignore\_failure": true

}

},

{

"geoip": {

"field": "data.gcp.jsonPayload.sourceIP",

"target\_field": "GeoLocation",

"properties": ["city\_name", "country\_name", "region\_name", "location"],

"ignore\_missing": true,

"ignore\_failure": true

}

},

{

"geoip": {

"field": "data.office365.ClientIP",

"target\_field": "GeoLocation",

"properties": ["city\_name", "country\_name", "region\_name", "location"],

"ignore\_missing": true,

"ignore\_failure": true

}

},

{

"date": {

"field": "timestamp",

"target\_field": "@timestamp",

"formats": ["ISO8601"],

"ignore\_failure": false

}

},

{

"date\_index\_name": {

"field": "timestamp",

"date\_rounding": "d",

"index\_name\_prefix": "{{fields.index\_prefix}}",

"index\_name\_format": "yyyy.MM.dd",

"ignore\_failure": false

}

},

{ "remove": { "field": "message", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "ecs", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "beat", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "input\_type", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "tags", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "count", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "@version", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "log", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "offset", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "type", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "host", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "fields", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "event", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "fileset", "ignore\_missing": true, "ignore\_failure": true } },

{ "remove": { "field": "service", "ignore\_missing": true, "ignore\_failure": true } }

],

"on\_failure" : [{

"drop" : { }

}]

}

Where the values:

M - stands for month

w - stands for week

d - stands for day

## The wazuh‑archives-\* indices

In addition to logging alerts to the /var/ossec/logs/alerts/alerts.json and /var/ossec/logs/alerts/alerts.log files, you can enable the Wazuh archives to log and index all the events the Wazuh server receives. This includes events that are analyzed by Wazuh and events that do not trigger alerts.

Storing and indexing all events might be useful for later analysis and compliance requirements. However, you must consider that enabling logging and indexing of all events will increase the storage requirement on the Wazuh server.

By default, the Wazuh indexer creates event indices for each unique day. You can modify the default index name in the /usr/share/filebeat/module/wazuh/archives/ingest/pipeline.json file of the Wazuh server. To do this:

1. Navigate to the date\_index\_name field.
2. Locate the date\_rounding key and change the default index name formatting in the /usr/share/filebeat/module/wazuh/archives/ingest/pipeline.json file.

The sections below provide details on how to enable the wazuh archives and set up the wazuh-archives-\* indices.

### Enabling Wazuh archives

1. Edit /var/ossec/etc/ossec.conf on the Wazuh server and set the <logall\_json> line to yes. This enables logging to [archives.json](https://documentation.wazuh.com/current/user-manual/reference/ossec-conf/global.html" \l "reference-ossec-global-logall-json) of all events. Forwarding to the Wazuh indexer requires the logging of all events in JSON format.
2. <logall\_json>yes</logall\_json>
3. Restart the Wazuh manager to make the change effective.
4. systemctl restart wazuh-manager

or

service wazuh-manager restart

1. Edit /etc/filebeat/filebeat.yml and change enabled to true in the archives mapping. This enables events to be forwarded to the Wazuh indexer.
2. filebeat.modules:
3. - module: wazuh
4. alerts:
5. enabled: true
6. archives:
7. enabled: true
8. Restart the Filebeat service to apply the change:
9. systemctl restart filebeat
10. Test that the Filebeat service works properly:
11. filebeat test output
12. elasticsearch: https://127.0.0.1:9200...
13. parse url... OK
14. connection...
15. parse host... OK
16. dns lookup... OK
17. addresses: 127.0.0.1
18. dial up... OK
19. TLS...
20. security: server's certificate chain verification is enabled
21. handshake... OK
22. TLS version: TLSv1.2
23. dial up... OK
24. talk to server... OK
25. version: 7.10.2

### Defining the index pattern

1. In the Wazuh dashboard upper left menu **☰**, go to **Dashboard management** > **Dashboard Management** and click **Index Patterns**.
2. Click on **Create index pattern**.
3. Set wazuh-archives-\* as the **Index pattern name**. This defines the index pattern to match the events being forwarded and indexed. Click on **Next step**.
4. Select **timestamp** for the **Time** field.

**Note**

Be careful to choose timestamp instead of @timestamp.

1. Click on **Create index pattern**.

### Viewing the index pattern

1. Click **Explore** on the upper left menu **☰**, and then click **Discover**.
2. Select **wazuh-archives-\*** to view the events.

## The wazuh-monitoring-\* indices

The connection status of an enrolled Wazuh agent at any moment is one of the following:

* **Active**
* **Disconnected**
* **Pending**
* **Never connected**

Wazuh stores a history of the connection status of all its agents. By default, it indexes the agent connection status using the wazuh‑monitoring-\* indices. The Wazuh indexer creates one of these indices per week by default. Check the documentation on [custom creation intervals](https://documentation.wazuh.com/current/user-manual/wazuh-dashboard/settings.html). These indices store the connection status of all the agents every 15 minutes by default. Check the documentation on the [frequency of API requests](https://documentation.wazuh.com/current/user-manual/wazuh-dashboard/settings.html).

The Wazuh dashboard requires these indices to display information about agent status. For example, by clicking **☰** > **Agents management** > **Summary**, you can see information such as the Wazuh agent's connection status and historical evolution within set timeframes.

In the [Wazuh dashboard configuration file](https://documentation.wazuh.com/current/user-manual/wazuh-dashboard/settings.html), you can change the settings to do the following:

* Disable inserting and showing connection status data for the agents. Change [wazuh.monitoring.enabled](https://documentation.wazuh.com/current/user-manual/wazuh-dashboard/settings.html) to accomplish this.
* Change the insertion frequency of connection status data for the agents. Change [wazuh.monitoring.frequency](https://documentation.wazuh.com/current/user-manual/wazuh-dashboard/settings.html) to accomplish this.

## The wazuh‑statistics-\* indices

The Wazuh dashboard uses the wazuh‑statistics-\* indices to display statistics about the Wazuh server usage and performance. The information displayed includes the number of events decoded, bytes received, and TCP sessions.

The Wazuh dashboard runs requests to the Wazuh manager API to query usage-related information. It inserts data into the wazuh‑statistics-\* indices from the information collected. The Wazuh indexer creates a wazuh‑statistics-\* index per week by default. Check the documentation on the [Statistics creation interval](https://documentation.wazuh.com/current/user-manual/wazuh-dashboard/settings.html). These indices store Wazuh server statistics every 5 minutes by default. Check the documentation on the [Frequency of task execution](https://documentation.wazuh.com/current/user-manual/wazuh-dashboard/settings.html).

To visualize this information in the Wazuh dashboard, go to **Server management** > **Statistics**.

## The wazuh-states-vulnerabilities-\* indices

The wazuh-states-vulnerabilities-\* index is used in Wazuh to store data related to the vulnerability state of monitored assets. This index typically contains information about vulnerabilities detected on monitored endpoints, including details such as the severity, status, affected software, and vulnerability reference. The \* at the end of the index pattern allows for the creation of multiple indices with similar names, segmented by time or other factors. This enables efficient storage and retrieval of vulnerability data over time.

To visualize this information in the Wazuh dashboard, click on **Vulnerability Detection** from the Wazuh dashboard home page.

## The wazuh-states-inventory-hardware-\* indices

The wazuh-states-inventory-hardware-\* index contains the baseline hardware inventory collected from monitored endpoints. Each document in this index represents details about the endpoint's hardware components, such as CPU and memory.

This index provides security teams and administrators with visibility into the underlying hardware of each endpoint. By tracking this information, Wazuh makes it possible to detect hardware changes, validate asset configurations, and support compliance or auditing efforts.

Since hardware details rarely change under normal circumstances, unexpected modifications captured in this index can serve as a signal for anomalies. For example, if you navigate to **IT Hygiene** > **System** > **Hardware**, you will see information relating to the hardware.

## The wazuh-states-inventory-hotfixes-\* indices

The wazuh-states-inventory-hotfixes-\* index stores information about Windows updates (hotfixes) installed on monitored endpoints. Each entry in this index corresponds to a specific update, including details such as the hotfix identifier, description, and installation date.

This index is closely related to the Wazuh vulnerability detection module. By cross-referencing installed hotfixes with known vulnerability databases, Wazuh can determine which vulnerabilities have already been patched and which ones remain unpatched on an endpoint.

Beyond vulnerability management, this index also helps administrators to verify system update compliance, audit patch history, and ensure that critical updates are consistently applied across their environment.

To find information relating to the hotfixes on Windows endpoints, navigate to **IT Hygiene** > **Software** > **Windows KBs**.

## The wazuh-states-inventory-interfaces-\* indices

The wazuh-states-inventory-interfaces-\* index stores detailed information about the network interfaces on monitored endpoints. Each document records attributes such as interface status (up or down), MAC address, and packet transfer statistics.

This index gives administrators visibility into the networking interfaces on their systems. By tracking interface activity and configuration changes, we detect unusual behavior such as interfaces going down unexpectedly, new interfaces appearing, or abnormal packet transfer patterns.

In addition to aiding security investigations, this index is also useful for operational monitoring, capacity planning, and verifying that network configurations remain consistent with organizational standards.

To find information relating to an endpoint's network interface, navigate to **IT Hygiene** > **Network** > **Interfaces**.

## The wazuh-states-inventory-networks-\* indices

The wazuh-states-inventory-networks-\* index contains information about the IPv4 and IPv6 addresses assigned to network interfaces on monitored endpoints. Each record shows details on the interface to IP address mapping, enabling visibility into how an endpoint is connected to the network.

This index is valuable for tracking changes in network configurations, such as new IP addresses being assigned and old ones being removed. Such changes can indicate legitimate reconfiguration, but they may also point to misconfigurations or suspicious activity.

By maintaining this inventory, Wazuh helps administrators validate network settings, support compliance requirements, and investigate security incidents where IP address history and assignments are needed.

To find information relating to an endpoint's IP address assignment, navigate to **IT Hygiene** > **Network** > **Addresses**.

## The wazuh-states-inventory-packages-\* indices

The wazuh-states-inventory-packages-\* index stores information about the software packages currently installed on monitored endpoints. Each record details a package and includes details such as the package name, version, and vendor.

This index provides the foundation for software inventory management within Wazuh. It also enables administrators to track changes in the software stack, verify compliance with organizational policies, and spot the presence of unauthorized or outdated applications.

This index is a critical component of Wazuh vulnerability detection, which cross-references package versions with known vulnerabilities to identify endpoints that may be exposed. In this way, the index not only aids asset management but also plays a direct role in improving security posture.

To find information relating to an endpoint's software packages, navigate to **IT Hygiene** > **Software** > **Packages**.

## The wazuh-states-inventory-ports-\* indices

The wazuh-states-inventory-ports-\* index records the open network ports detected on monitored endpoints. It includes details such as port numbers, associated services, and listening states.

By maintaining visibility into exposed ports, this index helps administrators identify unauthorized services, track changes in system exposure, and reduce the attack surface. Unsanctioned open ports can be an early sign of compromise or misconfiguration, making this data essential for both security monitoring and compliance audits.

To find information relating to ports on a monitored endpoint, navigate to **IT Hygiene** > **Network** > **Traffic**.

## The wazuh-states-inventory-processes-\* indices

The wazuh-states-inventory-processes-\* index contains information about the processes running on monitored endpoints. Each entry describes attributes like process name, PID (process ID), and an associated user.

Tracking running processes allows Wazuh to detect suspicious or unauthorized software execution. This visibility is crucial for spotting malware, persistence mechanisms, or rogue processes that may evade traditional defenses. It also provides administrators with a historical reference for system activity, supporting forensic investigations.

To find information relating to processes on a monitored endpoint, navigate to **IT Hygiene** > **Processes**.

## The wazuh-states-inventory-protocols-\* indices

The wazuh-states-inventory-protocols-\* index stores details about the network routing configuration and supported protocols for each network interface on monitored endpoints. This includes protocol types, routing tables, and interface associations.

Monitoring this information enables organizations to ensure network configurations align with expected baselines. Unexpected protocol changes or routing entries can indicate misconfigurations or malicious activity, such as traffic redirection or tunneling attempts.

To find information relating to protocols on a monitored endpoint, navigate to **IT Hygiene** > **Network** > **Protocols**.

## The wazuh-states-inventory-system-\* indices

The wazuh-states-inventory-system-\* index provides system-level details about each monitored endpoint, including the operating system, version, hostname, and architecture.

This index acts as a master record of core system attributes, supporting inventory management, compliance tracking, and security investigations. It also allows administrators to group and correlate alerts based on system name, OS type, and architecture.

To find system-level information relating to monitored endpoints, navigate to **IT Hygiene** > **System** > **OS**.

س-------------

# Re-indexing

When changes are made to the index’s data schema, it becomes necessary to re-index data to reflect these changes. Existing data may not match the updated schema without re-indexing, leading to data inconsistencies or errors during queries. Re-indexing lets you copy all or a subset of your data from a source index into a destination index.

To re-index an existing index, perform the following steps on either the Wazuh dashboard or the Wazuh server.

## Wazuh dashboard

1. Click on the **upper left menu ☰** and go to **Indexer management** then **Dev Tools**.
2. Enter the following API call, replacing my-source-index with the source index pattern and my-destination-index with the destination index pattern.
3. POST /\_reindex
4. {
5. "source":{
6. "index":"my-source-index"
7. },
8. "dest":{
9. "index":"my-destination-index"
10. }
11. }

For example:

POST /\_reindex

{

"source":{

"index":"wazuh-alerts-\*"

},

"dest":{

"index":"example-alerts"

}

}

**Output**

{

"took": 23655,

"timed\_out": false,

"total": 26849,

"updated": 0,

"created": 26849,

"deleted": 0,

"batches": 27,

"version\_conflicts": 0,

"noops": 0,

"retries": {

"bulk": 0,

"search": 0

},

"throttled\_millis": 0,

"requests\_per\_second": -1,

"throttled\_until\_millis": 0,

"failures": []

}

## Command line interface

Run the following command on any Wazuh central component that is allowed to authenticate to the Wazuh API. Replace <INDEXER\_USERNAME> and <INDEXER\_PASSWORD> with the indexer username and password:

curl -k -u "<INDEXER\_USERNAME>:<INDEXER\_PASSWORD>" -XPOST "https://<INDEXER\_IP\_ADDRESS>:9200/\_reindex" -H 'Content-Type: application/json' -d'

{

"source":{

"index":"my-source-index"

},

"dest":{

"index":"my-destination-index"

}

}'

For example:

root@wazuh-server:~$ curl -k -u "INDEXER\_USERNAME:INDEXER\_PASSWORD" -XPOST "https://<INDEXER\_IP\_ADDRESS>:9200/\_reindex" -H 'Content-Type: application/json' -d'

{

"source":{

"index":"wazuh-alerts-\*"

},

"dest":{

"index":"example-alerts"

}

}'

**Output**

{"took":18025,"timed\_out":false,"total":26854,"updated":26854,"created":0,"deleted":0,"batches":27,"version\_conflicts":0,"noops":0,"retries":{"bulk":0,"search":0},"throttled\_millis":0,"requests\_per\_second":-1.0,"throttled\_until\_millis":0,"failures":[]}

# Wazuh indexer tuning

This guide shows how to change settings to optimize the Wazuh indexer performance. To change the Wazuh indexer password, see the [Password management](https://documentation.wazuh.com/current/user-manual/user-administration/password-management.html) section.

* [Memory locking](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-tuning.html#memory-locking)
* [Shards and replicas](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/wazuh-indexer-tuning.html#shards-and-replicas)

## Memory locking

When the system is swapping memory, the Wazuh indexer may not work as expected. Therefore, it is important for the health of the Wazuh indexer node that none of the Java Virtual Machine (JVM) is ever swapped out to disk. To prevent any Wazuh indexer memory from being swapped out, configure the Wazuh indexer to lock the process address space into RAM as follows.

**Note**

You require root user privileges to run the commands described below.

1. Add the below line to the /etc/wazuh-indexer/opensearch.yml configuration file on the Wazuh indexer to enable memory locking:
2. bootstrap.memory\_lock: true
3. Modify the limit of system resources. Configuring system settings depends on the operating system of the Wazuh indexer installation.

SystemdSysVinit

* 1. Create a new directory for the file that specifies the system limits:
  2. mkdir -p /etc/systemd/system/wazuh-indexer.service.d/
  3. Run the following command to create the wazuh-indexer.conf file in the newly created directory with the new system limit added:
  4. cat > /etc/systemd/system/wazuh-indexer.service.d/wazuh-indexer.conf << EOF
  5. [Service]
  6. LimitMEMLOCK=infinity
  7. EOF

1. Edit the /etc/wazuh-indexer/jvm.options file and change the JVM flags. Set a Wazuh indexer heap size value to limit memory usage. JVM heap limits prevent the OutOfMemory exception if the Wazuh indexer tries to allocate more memory than is available due to the configuration in the previous step. The recommended value is half of the system RAM. For example, set the size as follows for a system with 8 GB of RAM.
2. -Xms4g
3. -Xmx4g

Where the total heap space:

* 1. -Xms4g - initial size is set to 4Gb of RAM.
  2. -Xmx4g - maximum size is to 4Gb of RAM.

**Warning**

To prevent performance degradation due to JVM heap resizing at runtime, the minimum (Xms) and maximum (Xmx) size values must be the same.

1. Restart the Wazuh indexer service:
2. systemctl daemon-reload
3. systemctl restart wazuh-indexer
4. Verify that the setting was changed successfully, by running the following command to check that mlockall value is set to true:
5. curl -k -u <INDEXER\_USERNAME>:<INDEXER\_PASSWORD> "https://<INDEXER\_IP\_ADDRESS>:9200/\_nodes?filter\_path=\*\*.mlockall&pretty"

**Output**

{

"nodes" : {

"sRuGbIQRRfC54wzwIHjJWQ" : {

"process" : {

"mlockall" : true

}

}

}

}

If the output is false, the request has failed, and the following line appears in the /var/log/wazuh-indexer/wazuh-indexer.log file:

Unable to lock JVM Memory

## Shards and replicas

The Wazuh indexer offers the possibility of splitting an index into multiple segments called shards. Each shard is a fully functional and independent "index" that can be hosted on any node in the Wazuh indexer cluster. The splitting is important for two main reasons:

* Horizontal scaling.
* Distribution and parallelization operations across shards, increasing performance and throughput.

In addition, the Wazuh indexer allows users to make one or more copies of the index shards in what are called replica shards, or replicas for short. Replication is important for two reasons:

* Provides high availability in case a shard or a node fails.
* Allows the search volume and throughput to scale since searches can be executed on all replicas in parallel.

### Number of shards for an index

Before creating the first index, consider carefully how many shards will be needed. It is not possible to change the number of shards without re-indexing.

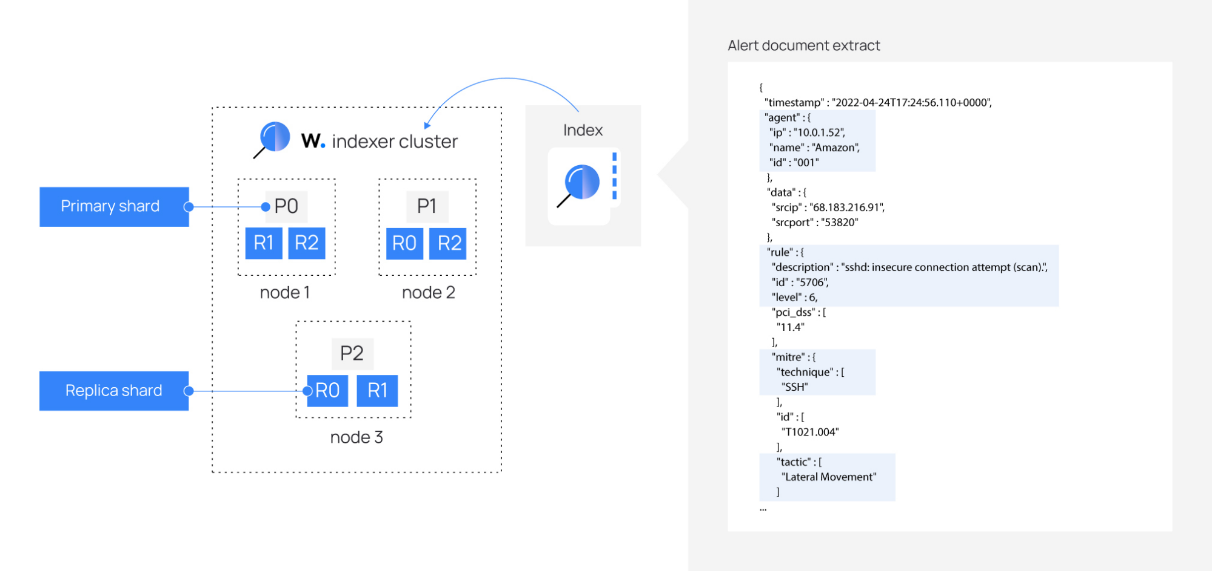
The number of shards needed for optimal performance depends on the number of nodes in the Wazuh indexer cluster. As a general rule, the number of shards must be the same as the number of nodes. For example, a cluster with three nodes should have three shards, while a cluster with only one node would only need one.

### Number of replicas for an index

The number of replicas depends on the available storage for the indices. Here is an example of how a Wazuh indexer cluster with three nodes and three shards could be set up.

* **No replica**: Each node has one shard. If one node goes down, an incomplete index of only two shards is available.
* **One replica**: Each node has one shard and one replica. If one node goes down, a full index is still available.
* **Two replicas**: Each node has the full index with one shard and two replicas. With this setup, the cluster continues to operate even if two nodes go down. Although this seems to be the best solution, it increases the storage requirements.

The image below shows a Wazuh indexer cluster with three nodes, each containing a primary shard and two replica shards.



### Setting the number of shards

**Warning**

The number of shards and replicas are defined per index at the time of index creation. Once the index is created, although the number of replicas can be changed dynamically, the number of shards cannot be changed without [re-indexing](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/re-indexing.html).

The default installation of a Wazuh indexer node creates each index with three primary shards and no replicas. You can modify the number of primary shards and replicas by loading a new template using the Wazuh API.

In the following example, we set the number of shards for a single-node Wazuh indexer to 1. Perform the following steps on the Wazuh indexer node or any central component allowed to authenticate using the Wazuh API.

1. Download the Wazuh indexer template:
2. curl https://raw.githubusercontent.com/wazuh/wazuh/v4.13.1/extensions/elasticsearch/7.x/wazuh-template.json -o w-indexer-template.json
3. Edit w-indexer-template.json to set index.number\_of\_shards to 1. To avoid Filebeat overwriting the existing template, set the order to 1. Multiple matching templates in the same order result in a non-deterministic merging order.
4. {
5. "order": 1,
6. "index\_patterns": [
7. "wazuh-alerts-4.x-\*",
8. "wazuh-archives-4.x-\*"
9. ],
10. "settings": {
11. "index.refresh\_interval": "5s",
12. "index.number\_of\_shards": "1",
13. "index.number\_of\_replicas": "0",
14. "index.auto\_expand\_replicas": "0-1",
15. "index.mapping.total\_fields.limit": 10000,
16. ...
17. Load the new settings.
18. curl -X PUT "https://<INDEXER\_IP\_ADDRESS>:9200/\_template/wazuh-custom" -H 'Content-Type: application/json' -d @w-indexer-template.json -k -u <INDEXER\_USERNAME>:<INDEXER\_PASSWORD>

**Output**

{"acknowledged":true}

1. Confirm that the configuration was successfully updated.
2. curl "https://<INDEXER\_IP\_ADDRESS>:9200/\_template/wazuh-custom?pretty&filter\_path=wazuh-custom.settings" -k -u <INDEXER\_USERNAME>:<INDEXER\_PASSWORD>

**Output**

{

"wazuh-custom" : {

"settings" : {

"index" : {

"mapping" : {

"total\_fields" : {

"limit" : "10000"

}

},

"refresh\_interval" : "5s",

"number\_of\_shards" : "1",

"auto\_expand\_replicas" : "0-1",

"number\_of\_replicas" : "0",

...

If the index has already been created, it must be [re-indexed](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/re-indexing.html).

### Setting the number of replicas

The number of replicas can be changed dynamically using the Wazuh indexer API. In a single-node cluster, the number of replicas should be set to zero. This is accomplished by running the following command on the Wazuh indexer node or any central component allowed to authenticate using the Wazuh API:

curl -k -u "<INDEXER\_USERNAME>:<INDEXER\_PASSWORD>" -XPUT "https://<INDEXER\_IP\_ADDRESS>:9200/wazuh-alerts-" -H 'Content-Type: application/json' -d'

{

"settings": {

"index": {

"number\_of\_replicas": 0

}

}

}'

# Migrating Wazuh indices

In this section, we focus on migrating Wazuh indices by using snapshots. This helps to restore alerts from one Wazuh indexer cluster to another without losing the original timestamp.

## Setup shared file system

We recommend the use of a Network File System (NFS) to create a shared file system for the snapshot repository.

### NFS server

Perform the following steps to set up NFS on a dedicated server:

1. Create a target directory for the snapshot repository in the /mnt directory:
2. mkdir /mnt/snapshots
3. Install NFS by running the following commands:

YumAPT

yum update

yum install -y nfs-utils

yum install exportfs

systemctl enable nfs-server

systemctl start nfs-server

1. Add the /mnt/snapshots directory to the /etc/exports file using the command below. Replace the <NETWORK\_ADDRESS/CIDR> variable with your network address.
2. echo "/mnt/snapshots <NETWORK\_ADDRESS/CIDR>(rw,sync,no\_root\_squash,no\_subtree\_check)" | sudo tee -a /etc/exports

Where:

* + rw - Allows both read and write access to the shared directory.
  + sync - Forces the NFS server to write changes to the disk immediately, making the file system synchronous.
  + no\_root\_squash - Allows the "root" user on the NFS client system to have full, unrestricted access to files on the NFS server.
  + no\_subtree\_check - Disables subtree checking, which can improve performance for large directory trees.

1. Apply the NFS configuration:
2. exportfs -a

### Wazuh indexer

Perform the following steps on the Wazuh indexer node (s) to complete the shared file system setup.

1. Create a target directory for the snapshot repository in the /mnt directory:
2. mkdir /mnt/snapshots
3. Install the NFS client:

YumAPT

yum -y install nfs-utils

1. Mount the shared directory /mnt/snapshots on the Wazuh indexer node(s). Replace the <NFS\_SERVER\_IP> variable with the IP address of the NFS server:
2. mount -t nfs <NFS\_SERVER\_IP>:/mnt/snapshots /mnt/snapshots
3. Grant the wazuh-indexer user ownership of the /mnt/snapshots directory:
4. chown wazuh-indexer:wazuh-indexer /mnt/snapshots
5. Add the configuration: path.repo: /mnt/snapshots to the /etc/wazuh-indexer/opensearch.yml file to specify the repository path:
6. network.host: "127.0.0.1"
7. node.name: "node-1"
8. cluster.initial\_master\_nodes:
9. - "node-1"
10. cluster.name: "wazuh-cluster"
11. node.max\_local\_storage\_nodes: "3"
12. path.data: /var/lib/wazuh-indexer
13. path.logs: /var/log/wazuh-indexer
14. path.repo: /mnt/snapshots
15. plugins.security.ssl.http.pemcert\_filepath: /etc/wazuh-indexer/certs/wazuh-indexer.pem
16. plugins.security.ssl.http.pemkey\_filepath: /etc/wazuh-indexer/certs/wazuh-indexer-key.pem
17. plugins.security.ssl.http.pemtrustedcas\_filepath: /etc/wazuh-indexer/certs/root-ca.pem
18. plugins.security.ssl.transport.pemcert\_filepath: /etc/wazuh-indexer/certs/wazuh-indexer.pem
19. plugins.security.ssl.transport.pemkey\_filepath: /etc/wazuh-indexer/certs/wazuh-indexer-key.>plugins.security.ssl.transport.pemtrustedcas\_filepath: /etc/wazuh-indexer/certs/root-ca.pem
20. plugins.security.ssl.http.enabled: true
21. plugins.security.ssl.transport.enforce\_hostname\_verification: false
22. plugins.security.ssl.transport.resolve\_hostname: false
23. plugins.security.ssl.http.enabled\_ciphers:
24. - "TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256"
25. - "TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384"
26. - "TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256"
27. - "TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384"
28. plugins.security.ssl.http.enabled\_protocols:
29. - "TLSv1.2"
30. plugins.security.authcz.admin\_dn:
31. - "CN=admin,OU=Wazuh,O=Wazuh,L=California,C=US"
32. plugins.security.check\_snapshot\_restore\_write\_privileges: true
33. plugins.security.enable\_snapshot\_restore\_privilege: true
34. plugins.security.nodes\_dn:
35. - "CN=indexer,OU=Wazuh,O=Wazuh,L=California,C=US"
36. plugins.security.restapi.roles\_enabled:
37. - "all\_access"
38. - "security\_rest\_api\_access"
39. plugins.security.system\_indices.enabled: true
40. plugins.security.system\_indices.indices: [".opendistro-alerting-config", ".opendistro-alert>
41. ### Option to allow Filebeat-oss 7.10.2 to work ###
42. compatibility.override\_main\_response\_version: true
43. Restart the Wazuh indexer to apply the configuration changes:
44. systemctl restart wazuh-indexer

**Warning**

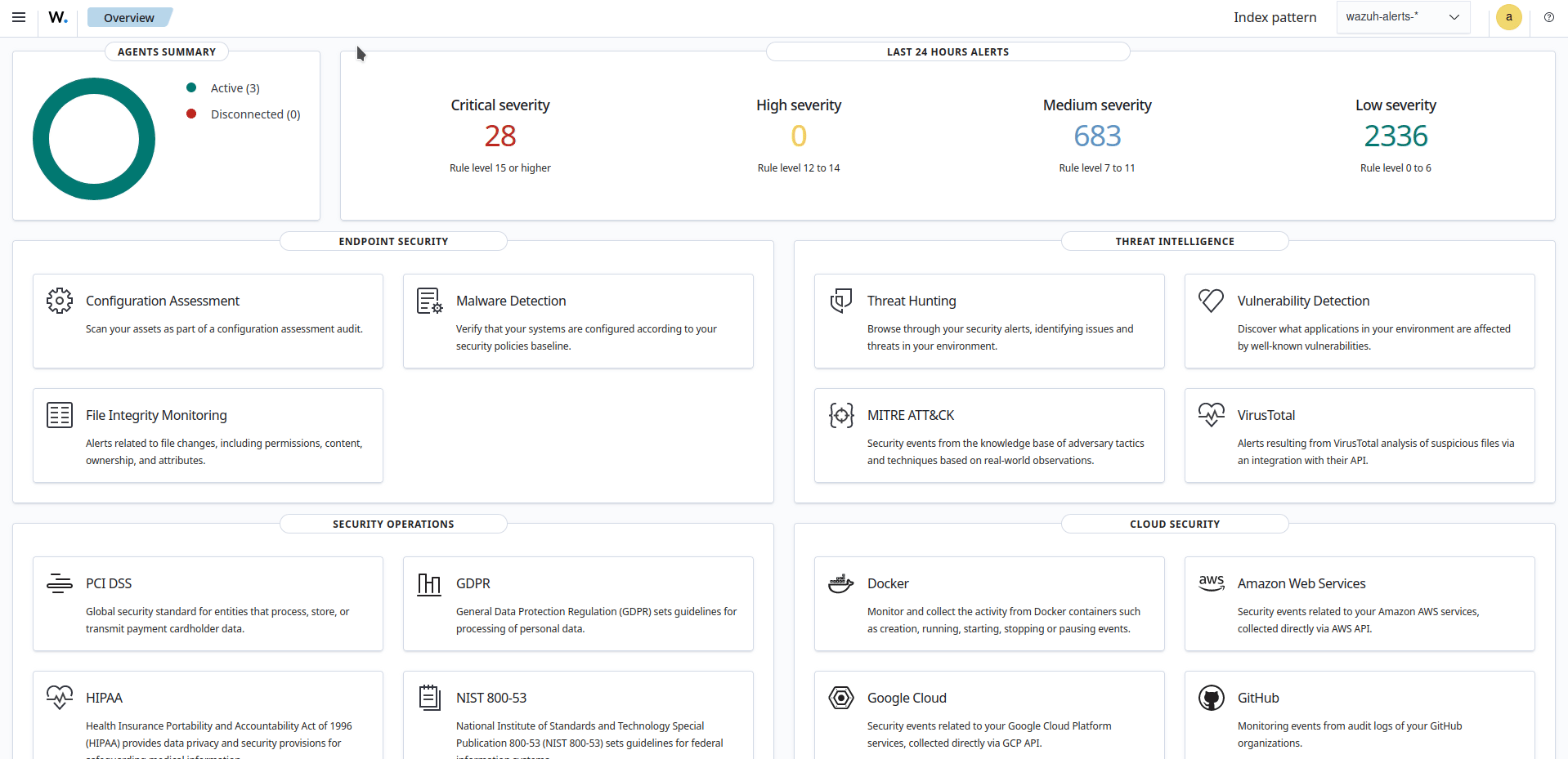
Make sure to confirm that the /mnt/snapshots directory has the wazuh-indexer:wazuh-indexer ownership on the Wazuh indexer nodes using the ll utility.

Repeat the [Setup shared file system > Wazuh indexer](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/migrating-wazuh-indices.html#migrating-indices-wazuh-indexer) steps on the destination Wazuh indexer(s) to use the NFS share directory, /mnt/snapshots, as its snapshot repository.

## Setup snapshot repository

On the Wazuh dashboard, perform the following steps:

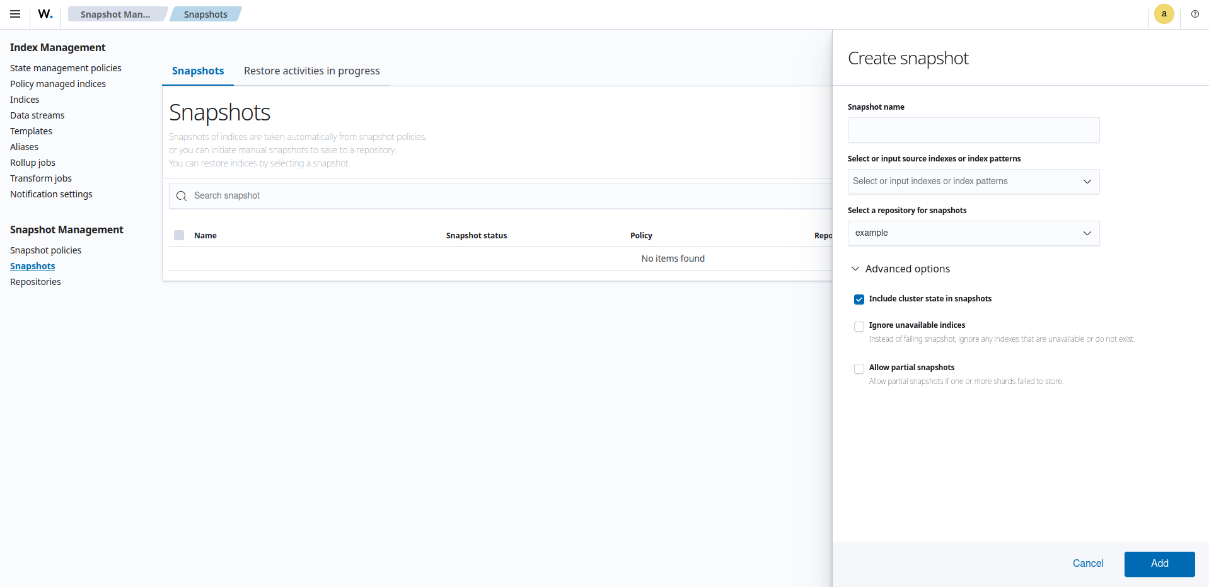
1. Click on the **upper left menu ☰**, go to **Indexer management** > **Snapshot Management** > **Repositories**, and select **Create repository** to create a new snapshot repository.
2. Enter a repository name, select the repository type **Shared file system**, enter the repository location /mnt/snapshots, and select **Add** to register the new repository.



Repeat the above steps on the destination Wazuh cluster to set up a similar snapshot repository.

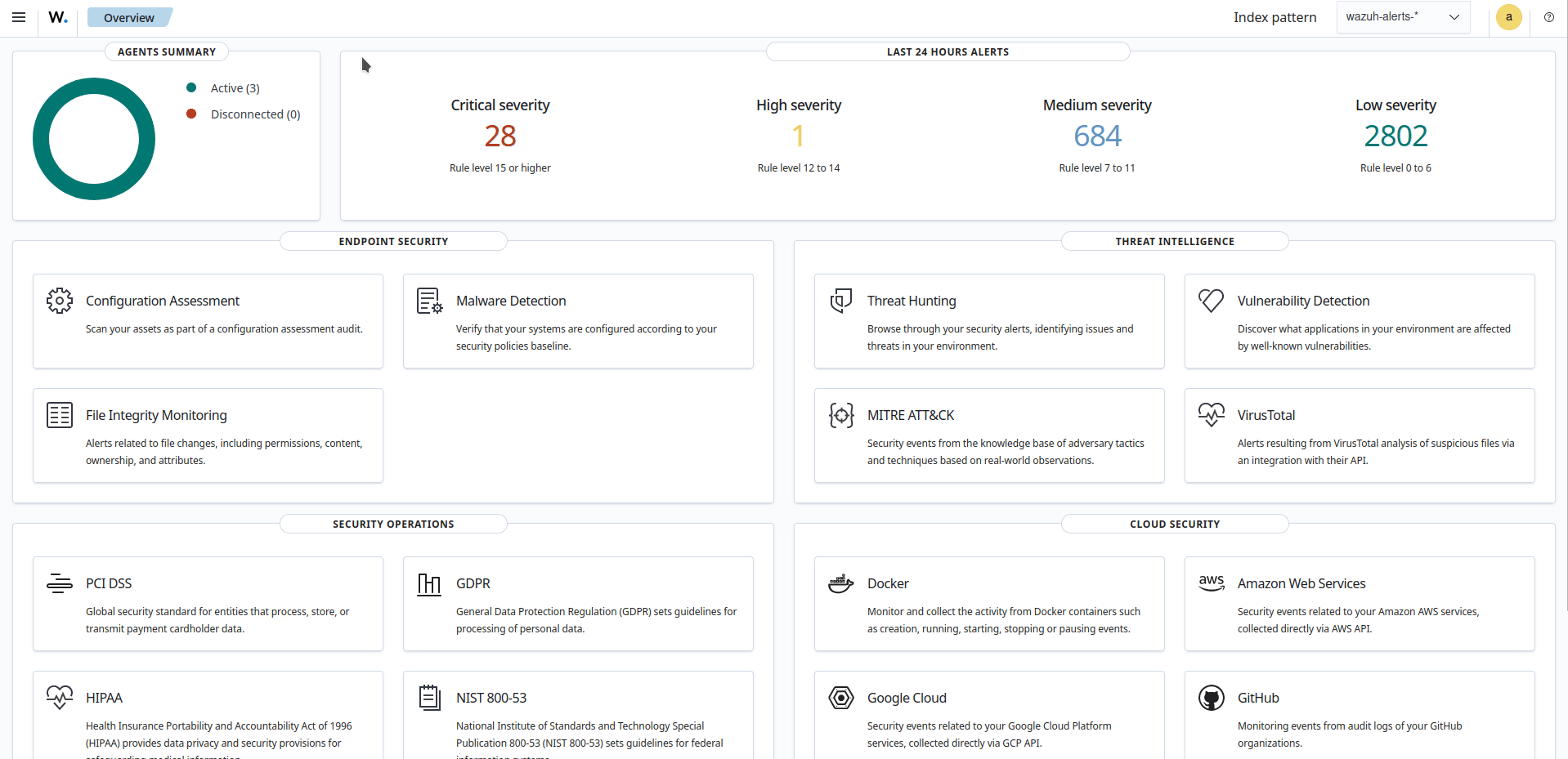
## Take snapshots

1. Click on the **upper left menu ☰**, and go to **Indexer management** > **Snapshot Management** > **Snapshots**.
2. Select **Take snapshot**, and enter a Snapshot name.
3. Select or input source index patterns.
4. Select the earlier created repository to store the snapshots.
5. Select **Advanced options** and check the **Include cluster state in snapshots** option.



1. Select **Add** to create a new snapshot.

The snapshot files are saved in the repository location /mnt/snapshots.



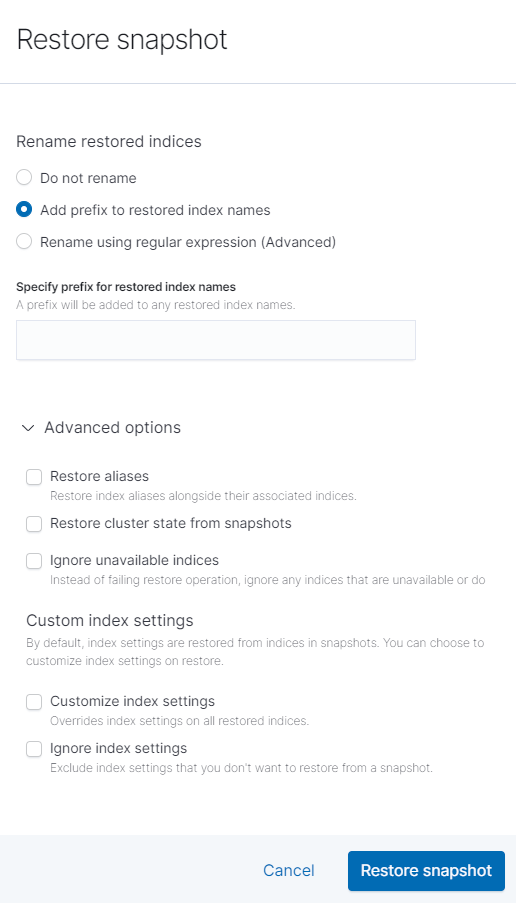
## Restore snapshots

To complete the Wazuh indices migration steps, restore the snapshots taken from the old Wazuh indexers to the destination Wazuh indexers. Perform the following steps on the destination Wazuh indexer.

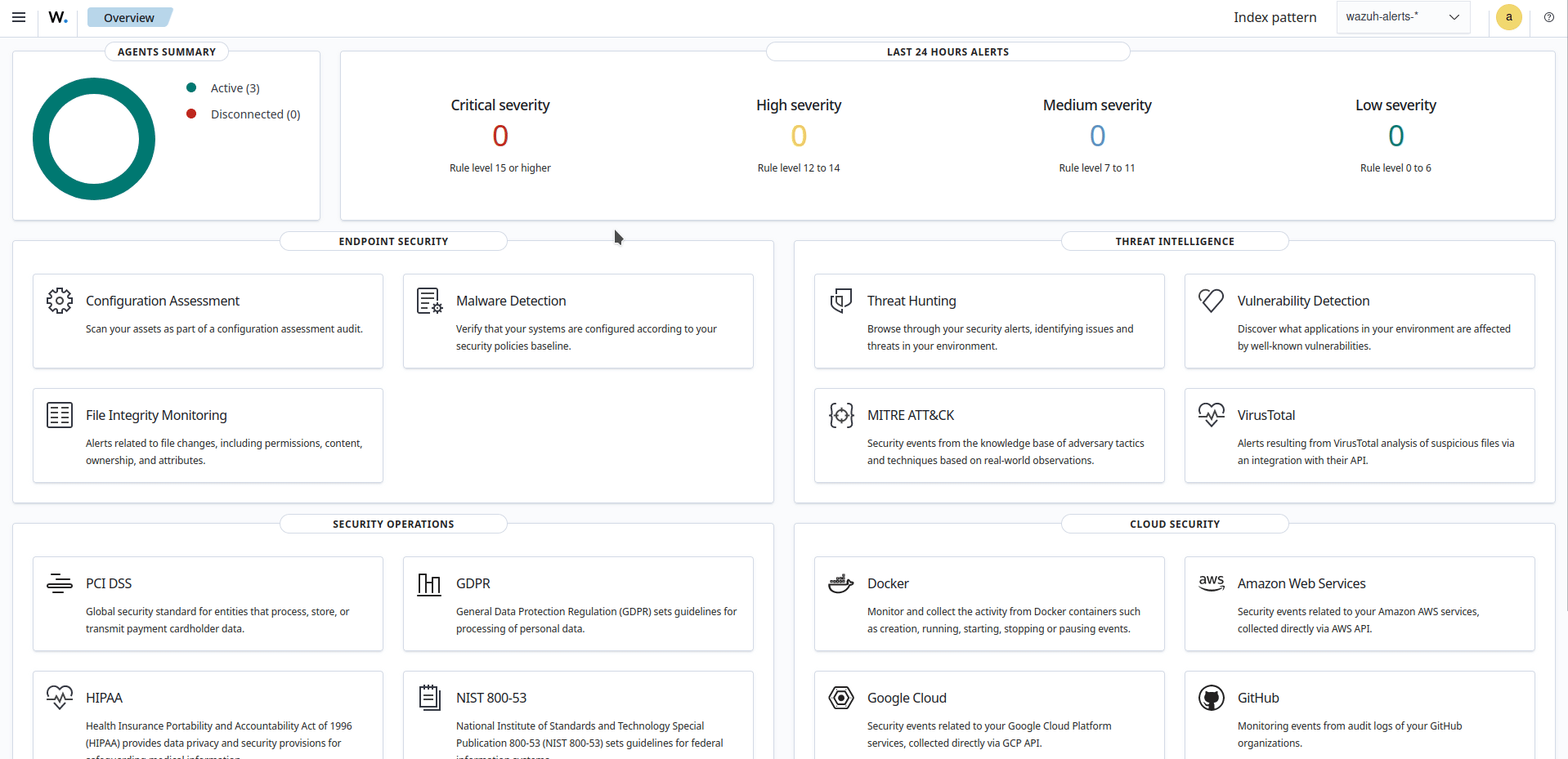
**Note**

It is necessary to have performed the steps in the [Setup shared file system](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/migrating-wazuh-indices.html#migrating-indices-setup-shared-file-system) and [Setup snapshot repository](https://documentation.wazuh.com/current/user-manual/wazuh-indexer/migrating-wazuh-indices.html#migrating-indices-setup-snapshot-repository) sections on the destination Wazuh cluster before proceeding to **Restore snapshots**.

1. Restart the Wazuh indexer nodes in the destination Wazuh cluster to load the snapshot files using the command:
2. systemctl restart wazuh-indexer
3. Click on the **upper left menu ☰**, go to **Indexer management** > **Snapshot Management** > **Snapshots**, and refresh the Snapshots page. The snapshots in the repository location /mnt/snapshots will show on the destination Wazuh cluster’s dashboard.
4. Select the snapshot and click on **Restore**. Delete the restored\_ prefix to restore the indices to their original names. The restored\_ prefix exists to avoid conflicting index names.
5. Select **Advanced options** and make sure all the options are unchecked.



1. Select **Restore snapshot** to complete the migration process.



# Wazuh indexer configuration on hardened endpoints

Wazuh indexer uses the Java Native Access (JNA) library for executing some platform-dependent native code. On Linux, the native code backing these libraries is extracted at runtime into a temporary directory and then mapped into executable pages in the indexer's address space. This requires the underlying files not to be on a filesystem mounted with the noexec option.

By default, the Wazuh indexer will create its temporary directory within /tmp. However, some hardened Linux installations mount /tmp with the noexec option by default. This prevents JNA from working correctly.

## Ensuring executable permissions in the Wazuh indexer temp directory

To resolve this problem, either remove the noexec option from your /tmp filesystem or configure the Wazuh indexer to use a different location. Follow the steps below to change the temporary directory of the Wazuh indexer by setting the $OPENSEARCH\_TMPDIR environment variable.

**Note**

You need root user privileges to run all the commands described below.

1. Create the temporary directory and set the appropriate permissions.
2. mkdir /var/lib/wazuh-indexer/tmp
3. chown wazuh-indexer:wazuh-indexer /var/lib/wazuh-indexer/tmp
4. Add Environment=OPENSEARCH\_TMPDIR=/var/lib/wazuh-indexer/tmp to the /lib/systemd/system/wazuh-indexer.service file, which is the systemd configuration file of the Wazuh indexer. The configuration file should be similar to the following:
5. [Service]
6. Type=notify
7. RuntimeDirectory=wazuh-indexer
8. PrivateTmp=true
9. Environment=OPENSEARCH\_HOME=/usr/share/wazuh-indexer
10. Environment=OPENSEARCH\_TMPDIR=/var/lib/wazuh-indexer/tmp Environment=OPENSEARCH\_PATH\_CONF=/etc/wazuh-indexer
11. Environment=PID\_DIR=/run/wazuh-indexer
12. Environment=OPENSEARCH\_SD\_NOTIFY=true
13. EnvironmentFile=-/etc/default/wazuh-indexer
14. Restart the Wazuh indexer service to apply the changes.
15. systemctl restart wazuh-indexer

## Handling unwanted Wazuh indexer restarts on Ubuntu

Modifying the Java temporary directory for the Wazuh indexer on some Ubuntu endpoints causes needrestart to detect normal Java operations in the directory as library changes. As a result, needrestart incorrectly flags the Wazuh indexer service as using outdated libraries and either prompts for a restart or automatically restarts the Wazuh indexer service. This occurs even when system package updates are unrelated to the Wazuh indexer. A workaround is to exclude the Wazuh indexer service from needrestart checks with the command below.

echo '$nrconf{blacklist\_rc} = [ qr(^wazuh-indexer) ];' > "/etc/needrestart/conf.d/wazuh-indexer.conf"

**Note**

This setting will make needrestart always ignore the Wazuh indexer service, even in cases where a restart would be legitimate. Therefore, users may choose to apply it at their discretion and risk.