[wazuh.com](https://www.google.com/imgres?imgurl=https://wazuh.com/wp-content/uploads/2017/01/Overview_general.png&imgrefurl=https://wazuh.com/&h=1329&w=1427&tbnid=AypO45eSyGcTPM:&tbnh=160&tbnw=171&usg=__1HI0FrbQwMVlv_hzPSsmA5nzjyc=&vet=1&docid=_8mhf7c76X8B8M&sa=X&ved=0ahUKEwjCho7LgfvVAhVpiFQKHd3dAZ0Q9QEIKjAA)

**Wazuh** is a free, open-source host-based intrusion detection system (HIDS). It performs log analysis, integrity checking, Windows registry monitoring, rootkit detection, time-based alerting, and active response.

Wazuh’s main components are the [agent](https://en.wikipedia.org/wiki/Software_agent) that runs on each monitored host, and the [server](https://en.wikipedia.org/wiki/Server_(computing)) that analyzes data received from the agents and from other agentless sources like [syslog](https://en.wikipedia.org/wiki/Syslog). In addition, the server forwards event data to an [Elasticsearch](https://en.wikipedia.org/wiki/Elasticsearch" \o "Elasticsearch) cluster, where information is indexed and stored.

* *Server*, the server component is the system that analyzes the data received from the agents, triggering alerts when an event matches a rule (e.g. [intrusion detection](https://en.wikipedia.org/wiki/Intrusion_detection), [file changed](https://en.wikipedia.org/w/index.php?title=File_changed&action=edit&redlink=1), configuration not compliant with policy, possible [rootkit](https://en.wikipedia.org/wiki/Rootkit), etc.).The server usually runs on a stand-alone physical server, [virtual machine](https://en.wikipedia.org/wiki/Virtual_machine), or cloud instance. This server typically will also run local agent components for the purpose of monitoring itself. Server runs only in [Linux](https://en.wikipedia.org/wiki/Linux).
* *Wazuh agent*, the Wazuh agent runs on monitored hosts that use a [Windows](https://en.wikipedia.org/wiki/Windows), [Linux](https://en.wikipedia.org/wiki/Linux), [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)), [BSD](https://en.wikipedia.org/wiki/BSD), or [Mac operating system](https://en.wikipedia.org/wiki/Mac_operating_system). It is used to detect security issues and collect different types of system and application data. The agent forwards the collected data to the Wazuh server through an encrypted and authenticated channel. In order to establish this secure channel, a registration process involving unique pre-shared keys is utilized.
* *Elastic Stack*, which is a unified suite of popular open source projects for log management, including [Elasticsearch](https://en.wikipedia.org/wiki/Elasticsearch" \o "Elasticsearch), [Logstash](https://en.wikipedia.org/wiki/Logstash" \o "Logstash), [Kibana](https://en.wikipedia.org/wiki/Kibana" \o "Kibana), [Filebeat](https://en.wikipedia.org/w/index.php?title=Filebeat&action=edit&redlink=1" \o "Filebeat (page does not exist)), and others. Wazuh integrates with [Elastic Stack](https://en.wikipedia.org/w/index.php?title=Elastic_Stack&action=edit&redlink=1) to provide a feed of already decoded log messages to be indexed by [Elasticsearch](https://en.wikipedia.org/wiki/Elasticsearch" \o "Elasticsearch), as well as a real-time web console for alert and log data analysis. In addition, Wazuh user interface (running on top of Kibana) can be used for management and monitoring of your Wazuh infrastructure.

It is compliant with [Payment Card Industry Data Security Standard](https://en.wikipedia.org/wiki/Payment_Card_Industry_Data_Security_Standard) (PCI DSS) requirements.

Distributed architectures do run the Wazuh server and Elastic Stack cluster (one or more servers) on different hosts. Single-host architectures have Wazuh server and Elastic Stack installed in the same system. This guide covers single-host architectures.

Installing Wazuh server

Adding Wazuh Repositories

The first thing you need is to add the Wazuh repository to your server. Alternatively, if you prefer to download the wazuh-manager package directly, you can find it [here](https://documentation.wazuh.com/current/installation-guide/packages-list/index.html#packages).

1. In order to perform this procedure properly, packages curl, apt-transport-https and lsb-release must be installed into your system. If they are not, install them:
2. $ apt-get update
3. $ apt-get install curl apt-transport-https lsb-release
4. Install the GPG key:
5. $ curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add -
6. Getting the distribution codename and adding the repository:
7. $ CODENAME**=$(**lsb\_release -cs**)**
8. $ echo "deb https://packages.wazuh.com/apt $CODENAME main" | tee /etc/apt/sources.list.d/wazuh.list

These are the supported codename values:

* For Debian: wheezy, jessie, stretch and sid
* For Ubuntu: trusty, vivid, wily, xenial and yakkety

1. Update the package information:
2. $ apt-get update

Installing Wazuh Manager

On your terminal, install the Wazuh manager:

$ apt-get install wazuh-manager

Once the process is completed, you can check the service status with:

1. For Systemd:
2. $ systemctl status wazuh-manager
3. For SysV Init:
4. $ service wazuh-manager status

Installing Wazuh API

1. NodeJS >= 4.6.1 is required in order to run the Wazuh API. If you do not have NodeJS installed, or your version is older than 4.6.1, we recommend you add the official NodeJS repository like this:
2. $ curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash -

and then, install nodejs:

$ apt-get install nodejs

1. Install the Wazuh API. It will update NodeJS if it is required:
2. $ apt-get install wazuh-api
3. Once the process is completed, you can check the service status with:
4. For Systemd:
5. $ systemctl status wazuh-api
6. For SysV Init:

$ service wazuh-api status

Installing Filebeat

Filebeat is the tool on the Wazuh server that will securely forward the alerts and archived events to the Logstash service on the Elastic Stack server(s).

In a single-host architecture as in our case (where Wazuh server and Elastic Stack are installed in the same system), you may entirely skip installing Filebeat, since Logstash will be able to read the event/alert data directly from the local filesystem without the assistance of a forwarder.

Once you have installed the manager, API and Filebeat (only needed for distributed architectures), you are ready to  install Elastic Stack.

Installing Elastic Stack

Many of the commands described below need to be executed with root user privileges.

Preparation

1. Oracle Java JRE is required by Logstash and Elasticsearch:
2. For Debian:

$ echo "deb http://ppa.launchpad.net/webupd8team/java/ubuntu xenial main" | tee /etc/apt/sources.list.d/webupd8team-java.list

$ echo "deb-src http://ppa.launchpad.net/webupd8team/java/ubuntu xenial main" | tee -a /etc/apt/sources.list.d/webupd8team-java.list

$ apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys EEA14886

1. For Ubuntu:

$ add-apt-repository ppa:webupd8team/java

1. Once the repository is added, install Java JRE:

$ apt-get update

$ apt-get install oracle-java8-installer

1. Install the Elastic repository and its GPG key:

$ apt-get install curl apt-transport-https

$ curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch | apt-key add -

$ echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | tee /etc/apt/sources.list.d/elastic-5.x.list

$ apt-get update

Elasticsearch

Elasticsearch is a highly scalable full-text search and analytics engine. For more info please see [Elasticsearch](https://www.elastic.co/products/elasticsearch).

1. Install the Elasticsearch package:

$ apt-get install elasticsearch

1. Enable and start the Elasticsearch service:
2. For Systemd:

$ systemctl daemon-reload

$ systemctl enable elasticsearch.service

$ systemctl start elasticsearch.service

1. For SysV Init:

$ update-rc.d elasticsearch defaults 95 10

$ service elasticsearch start

1. Load Wazuh Elasticsearch template:

$ curl https://raw.githubusercontent.com/wazuh/wazuh-kibana-app/master/server/startup/integration\_files/template\_file.json | curl -XPUT 'http://localhost:9200/\_template/wazuh' -H 'Content-Type: application/json' -d @-

1. Insert sample alert:

$ curl https://raw.githubusercontent.com/wazuh/wazuh-kibana-app/master/server/startup/integration\_files/alert\_sample.json | curl -XPUT "http://localhost:9200/wazuh-alerts-"`date +%Y.%m.%d`"/wazuh/sample" -H 'Content-Type: application/json' -d @-

It is recommended to edit the default configuration to improve the Elasticsearch performance. To do so, please see Appendix [Elasticsearch tuning](https://documentation.wazuh.com/current/installation-guide/optional-configurations/elastic-tuning.html#elastic-tuning).

Logstash

Logstash is the tool that will collect, parse, and forward to Elasticsearch for indexing and storage all logs generated by Wazuh server. For more info please see [Logstash](https://www.elastic.co/products/logstash).

1. Install the Logstash package:

$ apt-get install logstash

1. Download the Wazuh config and template files for Logstash:

$ curl -so /etc/logstash/conf.d/01-wazuh.conf https://raw.githubusercontent.com/wazuh/wazuh/2.0/extensions/logstash/01-wazuh.conf

$ curl -so /etc/logstash/wazuh-elastic5-template.json https://raw.githubusercontent.com/wazuh/wazuh/2.0/extensions/elasticsearch/wazuh-elastic5-template.json

1. **Follow this step only if you are using a single-host architecture**:
2. Edit /etc/logstash/conf.d/01-wazuh.conf, commenting out the entire input section titled “Remote Wazuh Manager - Filebeat input” and uncommenting the entire input section titled “Local Wazuh Manager - JSON file input”. This will set up Logstash to read the Wazuh alerts.json file directly from the local filesystem rather than expecting Filebeat on a separate server to forward the information in that file to Logstash.
3. Because the Logstash user needs to read alerts.json file, please add it to OSSEC group by running:

$ usermod -a -G ossec logstash

1. Enable and start the Logstash service:
2. For Systemd:

$ systemctl daemon-reload

$ systemctl enable logstash.service

$ systemctl start logstash.service

1. For SysV Init:

$ update-rc.d logstash defaults 95 10

$ service logstash start

**Note**

If you are running Wazuh server and the Elastic Stack server on separate systems (**distributed architecture**), then it is important to configure encryption between Filebeat and Logstash. To do so, please see [Setting up SSL for Filebeat and Logstash](https://documentation.wazuh.com/current/installation-guide/optional-configurations/elastic_ssl.html#elastic-ssl).

Kibana

Kibana is a flexible and intuitive web interface for mining and visualizing the events and archives stored in Elasticsearch. More info at [Kibana](https://www.elastic.co/products/kibana).

1. Install the Kibana package:

$ apt-get install kibana

1. Install the Wazuh App plugin for Kibana:

$ /usr/share/kibana/bin/kibana-plugin install https://packages.wazuh.com/wazuhapp/wazuhapp.zip

**Warning**

The Kibana plugin installation process may take several minutes. Please wait patiently.

1. **Optional.** Kibana will listen only on the loopback interface (localhost) by default. To set up Kibana to listen on all interfaces, edit the file /etc/kibana/kibana.yml. Uncomment the setting server.host and change the value to:

server.host: "0.0.0.0"

**Note**

It is recommended to set up an Nginx proxy for Kibana in order to use SSL encryption and to enable authentication. Instructions to set the proxy up can be found at [Setting up SSL and authentication for Kibana](https://documentation.wazuh.com/current/installation-guide/optional-configurations/kibana_ssl.html#kibana-ssl).

1. Enable and start the Kibana service:
2. For Systemd:

$ systemctl daemon-reload

$ systemctl enable kibana.service

$ systemctl start kibana.service

1. For SysV Init:

$ update-rc.d kibana defaults 95 10

$ service kibana start

Connecting the Wazuh App with the API

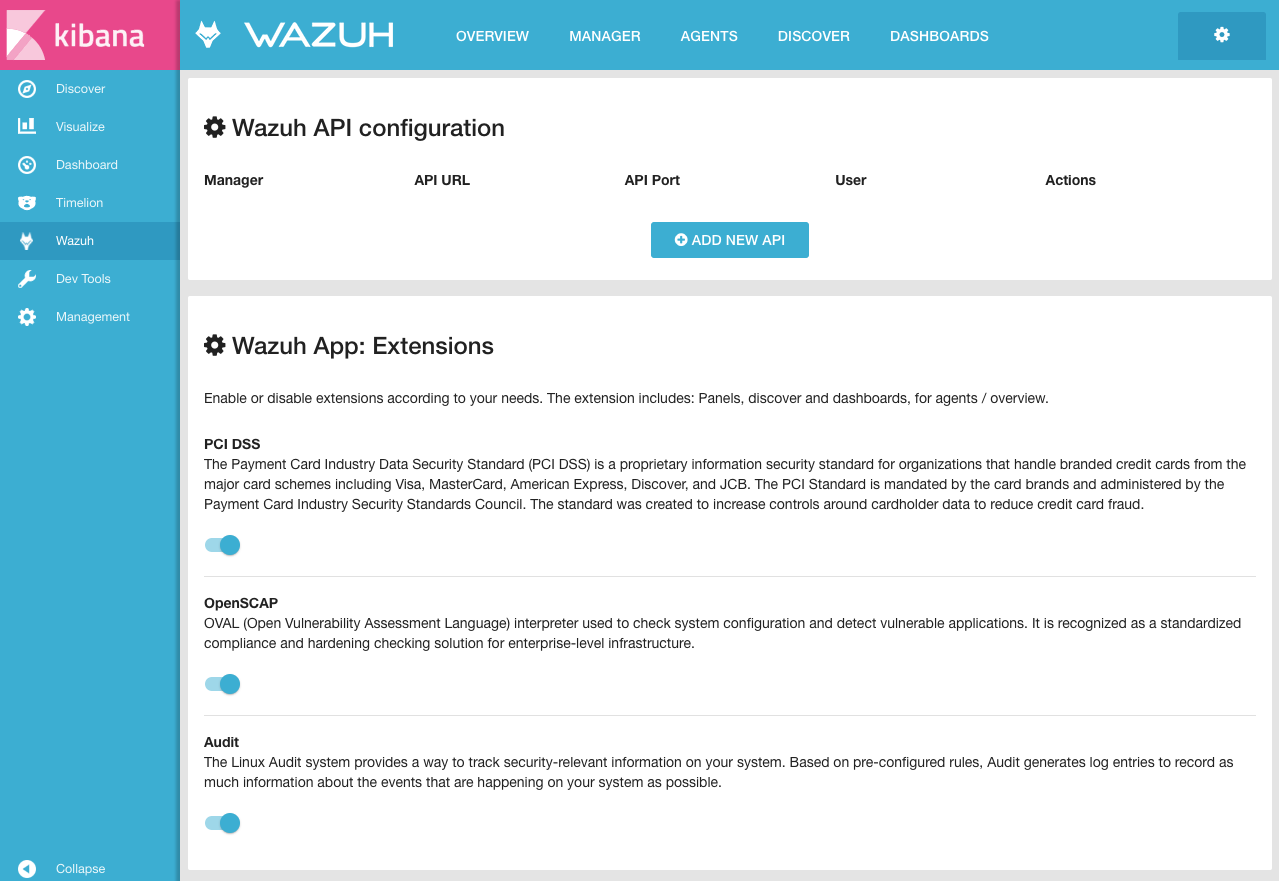
Follow the next guide in order to connect the Wazuh App with the API:

* [Connect the Wazuh App with the API](https://documentation.wazuh.com/current/installation-guide/installing-elastic-stack/connect_wazuh_app.html)

Connect the Wazuh App with the API

In this section, we’ll register the Wazuh API (installed on the Wazuh server) into the Wazuh App in Kibana:

1. Open a web browser and go to the Elastic Stack server’s IP address on port 5601 (default Kibana port). Then, from the left menu, go to the Wazuh App.
2. Click on Add new API.

[](https://documentation.wazuh.com/current/_images/connect_api.png)

1. Before filling out the fields, go to your Wazuh server and using the command prompt as root set a non-default credentials to protect your Wazuh API:

*# Replace your desired username for myUserName.*

$ cd /var/ossec/api/configuration/auth

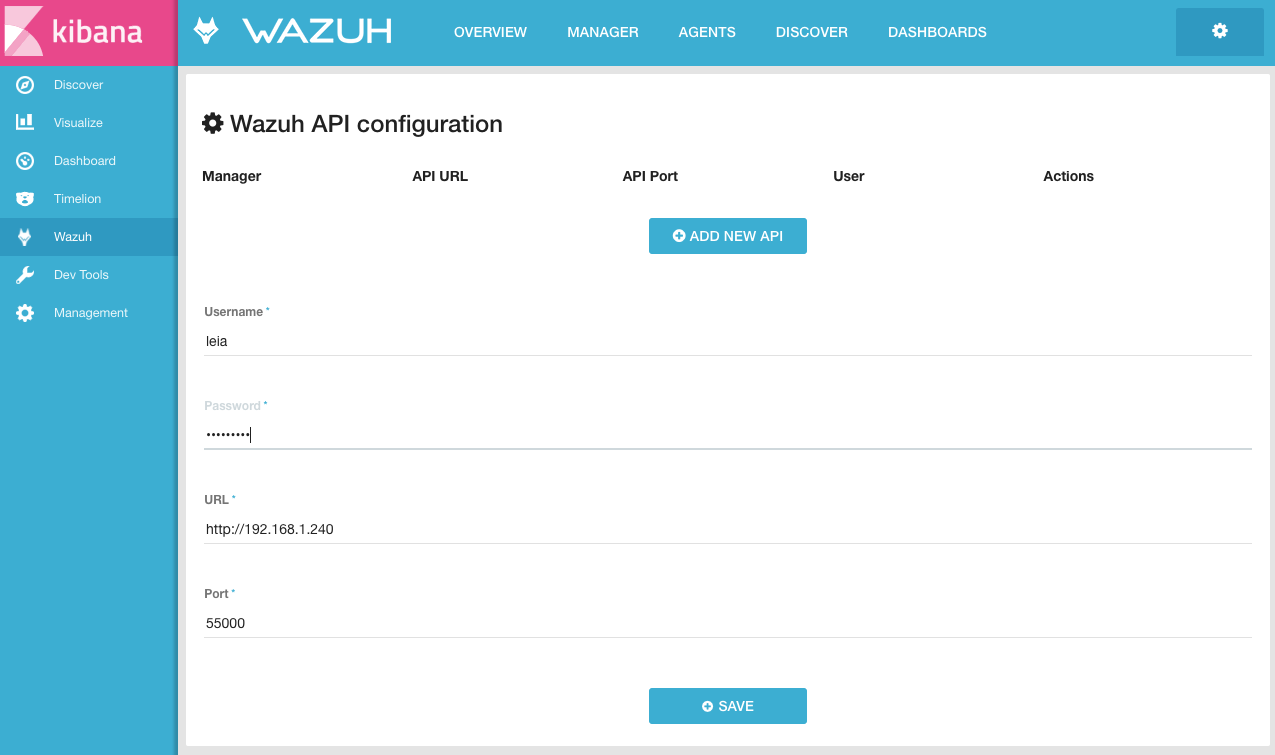
$ sudo node htpasswd -c user myUserName

*# Do not forget to restart the API to apply the changes:*

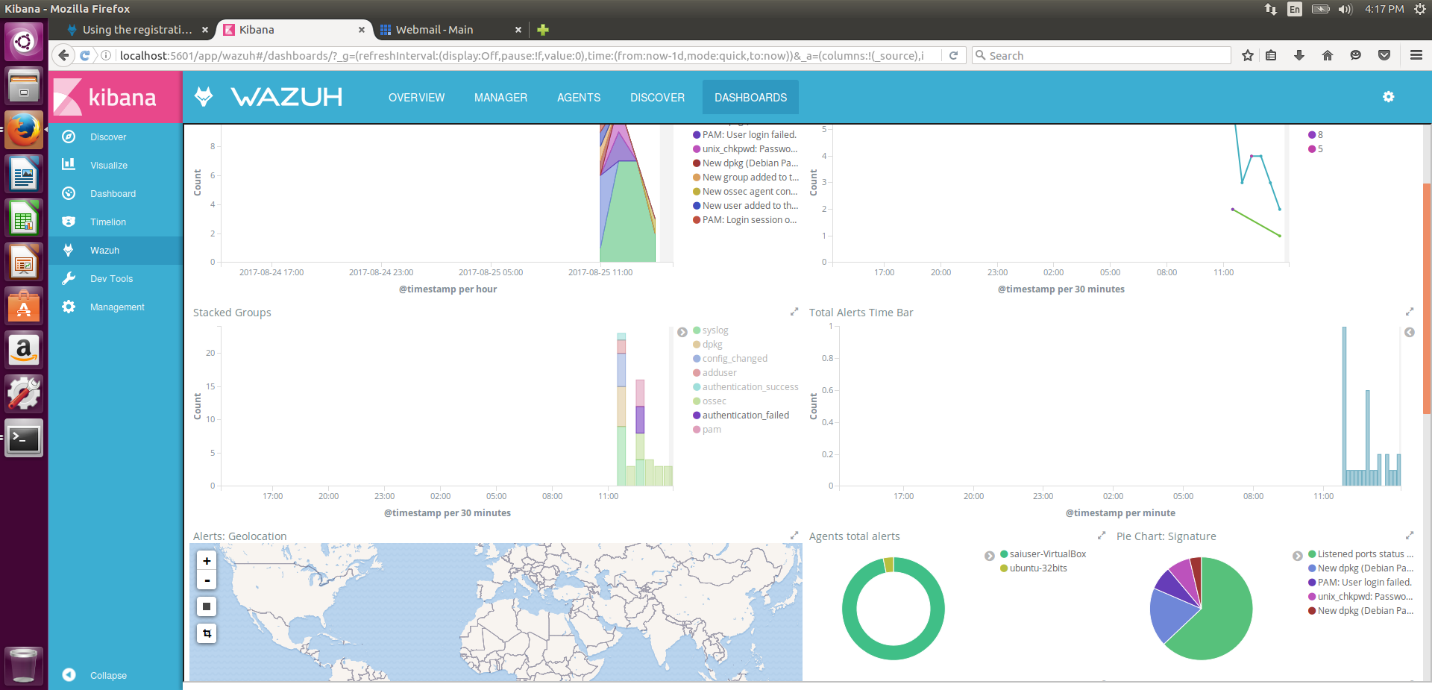
$ systemctl restart wazuh-api

$ service wazuh-api restart

1. Fill Username and Password with appropriate credentials you created in previous step. Enter http://MANAGER\_IP for the URL, where MANAGER\_IP is the real IP address of the Wazuh qserver. Enter “55000” for the Port.

[](https://documentation.wazuh.com/current/_images/fields_api.png)

1. Click on Save.



Once the Wazuh and Elastic Stack servers are installed and connected, you can install and connect Wazuh agents.

The Wazuh agent runs on the hosts that you want to monitor. It is multi-platform and provides the following capabilities: log and data collection, file integrity monitoring, rootkits and malware detection, and security policy monitoring. In addition, it talks to the Wazuh manager, sending data in near real-time through an encrypted and authenticated channel.

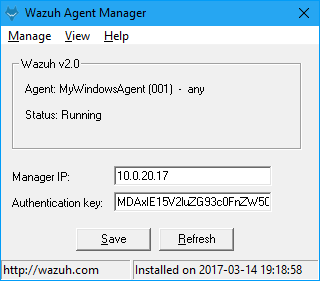
We installed Agent Windows server and in Unbuntu .

Install Wazuh agent on Windows

Download the Windows installer from:

https://documentation.wazuh.com/current/installation-guide/packages-list/index.html

Using the GUI:



By default all agent files can be found at the following location: C:\Program Files(x86)\ossec-agent.

Manager IP: Is the address of the Server manager

The Authentication Key: we import ( past) the key we already export when we run the manage\_agents In the server manager side.

At this point your agent is installed and you just need to register and configure it to talk to your manager

The Agent need to be running as local service:

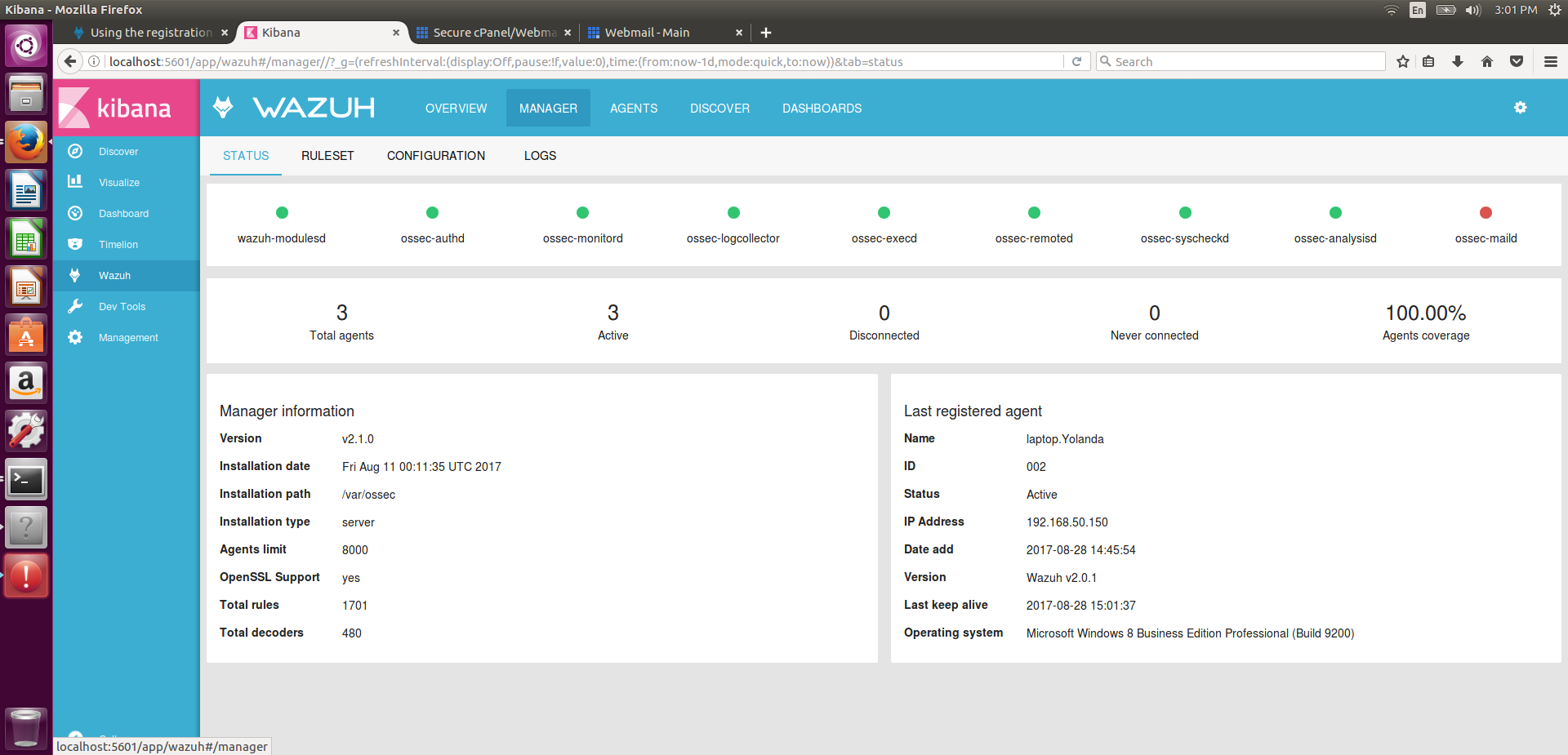
Service Name:OssecSvc

Display Name:Wazuh

Description: Wazuh Windows Agent.

The manager IP is the IP of the Wazuh server

Authentication Key is the key that need to be exported and imported in this agent in the process of registration the agent.



Install Wazuh agent with DEB packages

The DEB package is suitable for Debian, Ubuntu, and other Debian-based systems.

Many of the commands described below need to be executed with root user privileges.

Adding the Wazuh repository

The first thing you need is to add the Wazuh repository to your server. Alternatively, if you prefer to download the wazuh-agent package directly, you can find it [here](https://documentation.wazuh.com/current/installation-guide/packages-list/index.html#packages).

1. In order to perform this procedure properly, packages curl, apt-transport-https and lsb-release must be present on your system. If they are not, install them:
2. $ apt-get install curl apt-transport-https lsb-release
3. Install the Wazuh repository GPG key:
4. $ curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add -
5. Getting the distribution codename and adding the repository:
6. CODENAME**=$(**lsb\_release -cs**)**
7. echo "deb https://packages.wazuh.com/apt $CODENAME main" \
8. | tee /etc/apt/sources.list.d/wazuh.list

These are the supported codename values:

* For Debian: wheezy, jessie, stretch and sid
* For Ubuntu: trusty, vivid, wily, xenial and yakkety

1. Update the package information:
2. $ apt-get update

Installing Wazuh agent

On your terminal, install the Wazuh agent:

$ apt-get install wazuh-agent

At this point your agent is installed and you just need to register and configure it to talk to your manager.

The way we use in this installation was with the utility:

manage\_agents

The manage\_agents program is available in both a version for server and agent installations. The purpose of manage\_agents is to provide an easy-to-use interface to handle authentication keys for Wazuh agents. These authentication keys are required for secure (encrypted and authenticated) communication between the Wazuh server and its affiliated agent instances.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Wazuh v2.1.0 Agent manager. \*

\* The following options are available: \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(I)mport key from the server (I).

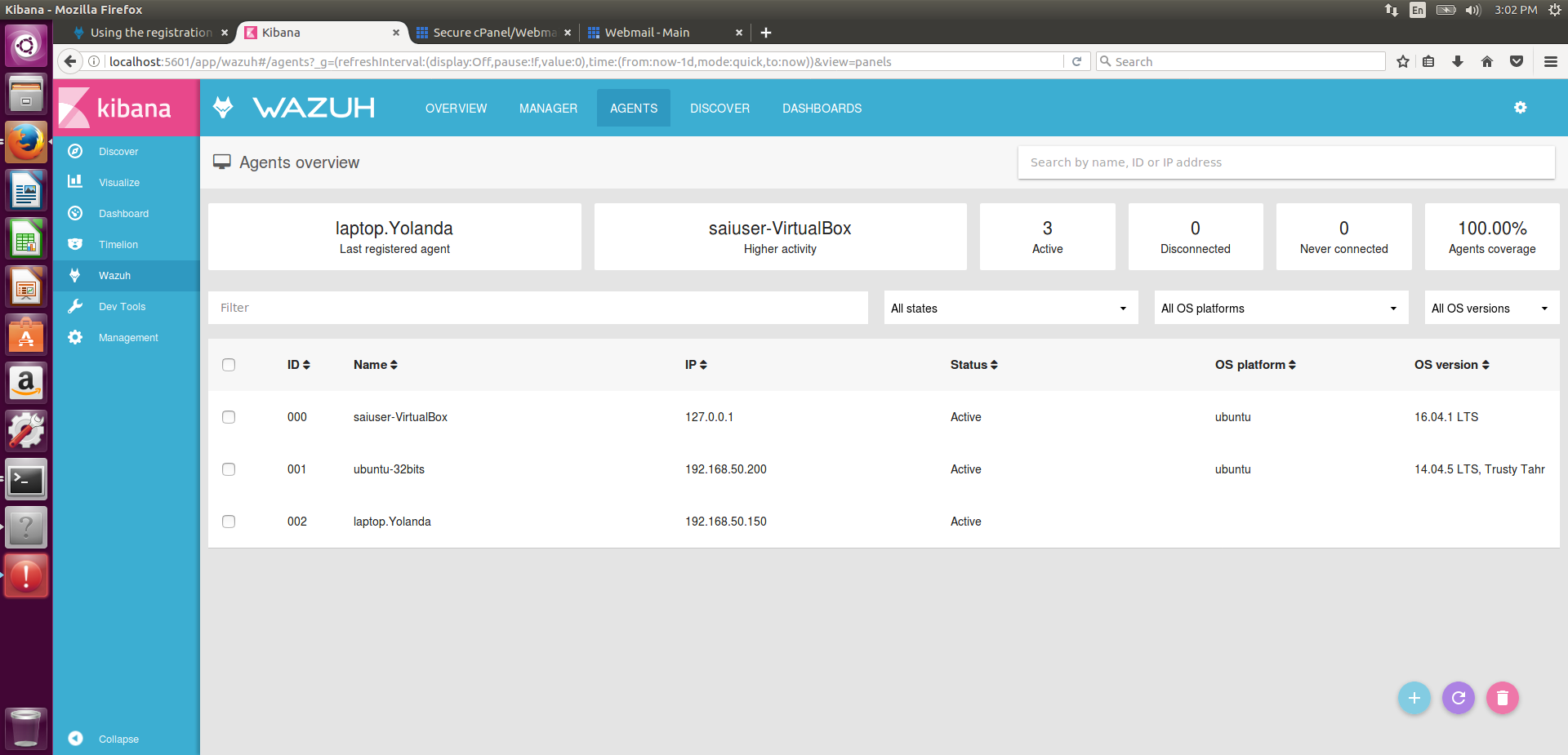
(Q)uit.

Choose your action: I or Q:

We chose I to import the key we already export when we run the manage\_agents

In the server manager side.

After we install both agents, we can see those by kibana interface:



Appendix:

Elasticsearch tuning

This guide summarizes the relevant configurations that allow us to optimize Elasticsearch.

1. [Memory locking](https://documentation.wazuh.com/current/installation-guide/optional-configurations/elastic-tuning.html#memory-locking)
2. [Shards and replicas](https://documentation.wazuh.com/current/installation-guide/optional-configurations/elastic-tuning.html#shards-and-replicas)

Memory locking

Elasticsearch performs poorly when the system is swapping the memory. It is vitally important to the health of your node that none of the JVM is ever swapped out to disk.

We will set the *bootstrap.memory\_lock* setting to true, so Elasticsearch will lock the process address space into RAM, preventing any Elasticsearch memory from being swapped out.

Step 1: Set bootstrap.memory\_lock

Uncomment or add this line to the /etc/elasticsearch/elasticsearch.yml file:

bootstrap.memory\_lock: true

Step 2: Edit limit of system resources

Where to configure systems settings depends on which package and operating system you choose to use for Elasticsearch installation.

* In a case where **systemd** is used, system limits need to be specified via systemd. First, create the folder executing the command: mkdir -p /etc/systemd/system/elasticsearch.service.d/, add a file called elasticsearch.conf and specify any changes in that file:

**[Service]**

LimitMEMLOCK**=**infinity

* In other case, edit the proper file /etc/sysconfig/elasticsearch for RPM or /etc/default/elasticsearch for Debian:

MAX\_LOCKED\_MEMORY**=**unlimited

Step 3: Limit memory

The previous configuration might cause node instability or even node death (with an *OutOfMemory* exception) if Elasticsearch tries to allocate more memory than is available. JVM heap limits will help us to define the memory usage and prevent this situation.

There are two rules to apply when setting the Elasticsearch heap size:

* No more than 50% of available RAM.
* No more than 32 GB.

By default, Elasticsearch is configured with a 1 GB heap. You can change the heap size via JVM flags using the /etc/elasticsearch/jvm.options file:

*# Xms represents the initial size of total heap space*

*# Xmx represents the maximum size of total heap space*

-Xms4g

-Xmx4g

Ensure that the min (Xms) and max (Xmx) sizes are the same, this prevents JVM heap resizing at runtime, a very costly process.