

# High Availability

High Availability is a feature of the [Data Center Edition](#).

## Description

Once the [the SonarQube cluster is installed](#), you have a High Availability configuration that will allow your SonarQube instance to stay up and running even if there is a crash or failure in one of the nodes of the cluster.

- [Description](#)
- [Start/Stop/Upgrade the Cluster](#)
  - [Start the Cluster](#)
  - [Stop the Cluster](#)
  - [Upgrade SonarQube](#)
- [Install/Upgrade a Plugin](#)
- [Monitoring](#)
  - [Manually Check the Status of your SQ Cluster from the UI](#)
- [Compute Engine Workers](#)
- [Project Move](#)
- [Configuration details](#)
- [Frequently Asked Questions](#)
- [Limitations](#)

## Start/Stop/Upgrade the Cluster

### Start the Cluster

1. Start the search nodes
2. Start the application nodes

### Stop the Cluster

1. Stop the application nodes
2. Stop the search nodes

### Upgrade SonarQube

1. Stop the cluster
2. Upgrade SonarQube on all nodes (app part, plugins, JDBC driver if required) following the usual [Upgrade](#) procedure but without triggering the `/set up` phase
3. Once all nodes have the same binaries: start the cluster
4. At this point only one of the application nodes is up. Try to access `[node ip:port]/setup` on each server, and trigger the setup operation on the one that responds.

## Install/Upgrade a Plugin

1. Stop the cluster
2. Upgrade the plugin on all nodes
3. Start the cluster

## Monitoring

CPU and RAM usage on each node have to be monitored separately with an APM.

In addition, we provide a Web API `api/system/health` you can use to validate all of the nodes of your cluster are operation.

- GREEN: SonarQube is fully operational
- YELLOW: SonarQube is usable, but it needs attention in order to be fully operational
- RED: SonarQube is not operational

To call it from monitoring system without having to give admin credentials, it is possible to setup a System Passcode through the property `sonar.web.systemPasscode`. This has to be configured in the `sonar.properties`.

### Manually Check the Status of your SQ Cluster from the UI

In the System Info page, you can check whether your cluster is running safely (green) or has some nodes with problems (orange or red).

# Compute Engine Workers

If you change the [number of Compute Engine workers](#) in the UI, you must restart each application node to have it take this change into account.

## Project Move

When the [Project Move](#) feature is used in a DC installation:

- Projects are exported on only one of the application nodes
- The archive of the exported projects must be copied to all the applications nodes in the target server

## Configuration details

(follow the [default configuration](#) first, details below is only for specific cases)

[Hazelcast](#) is used to manage the communication between the nodes of the cluster. You don't need to install it yourself, it's provided out of the box.

The following properties may be defined in the \$SONARQUBE\_HOME/config/sonar.properties file of each node in a cluster. When defining a property that contains a list of hosts (\* .hosts) the port is not required if the default port was not overridden in the configuration.

Property	Description	Default	Mandatory	Value for application nodes	Value for search nodes
sonar.cluster.enabled	Activates the cluster mode	false	mandatory	true	true
sonar.cluster.name	The name of the cluster. <b>Required if multiple clusters are present on the same network.</b> For example this prevents mixing Production and Preproduction clusters.  This will be the name stored in the Hazelcast cluster and used as the name of the Elasticsearch cluster.	sonarqube	optional	(see description)	(see description)
sonar.cluster.hosts	Comma-delimited list of all hosts in the cluster. Items must contain the port if the default <i>sonar.cluster.node.port</i> value is not used. Items format is <i>sonar.cluster.node.host</i> or <i>sonar.cluster.node.host:sonar.cluster.node.port</i> .  <b>FROM DATA CENTER EDITION 7.2+</b> : <i>sonar.cluster.hosts</i> must contain only application hosts.	(none)	mandatory	(see description)	(see description)
sonar.cluster.search.hosts	Comma-delimited list of search hosts in the cluster. Items must contain the port if the default <i>sonar.search.port</i> value is not used. Items format is <i>sonar.search.host</i> or <i>sonar.search.host:sonar.search.port</i> .	(none)	mandatory	(see description)	(see description)
sonar.cluster.node.name	The name of the node that is used on Elasticsearch and stored in Hazelcast member attribute (NODE_NAME) for sonar-application	sonarqube-{UUID}	optional	(see description)	(see description)
sonar.cluster.node.type	Type of node: either <b>application</b> or <b>search</b>	(none)	mandatory	application	search
sonar.cluster.node.host	<b>IP address</b> of the network card that will be used by Hazelcast to communicate with the members of the cluster.  <b>If not specified, the first interface will be chosen (note that loopback interfaces won't be selected)</b>	(none)	optional	(see description)	(see description)
sonar.cluster.node.port	The Hazelcast port for communication with each host (member) of the cluster.  <b>FROM DATA CENTER EDITION 7.2+</b> : <i>sonar.cluster.node.port</i> must be defined only application nodes.	9003	optional	(see description)	(see description)  From 7.2+: none
sonar.cluster.node.web.port	Hazelcast port for communication with the ComputeEngine process, only for application nodes. Port must be accessible to all other search and application nodes.  <b>If not specified, a dynamic port will be chosen and all ports must be open among the nodes.</b>	(none)	optional	(see description)	(none)

sonar. cluster. node.ce. port	Hazelcast port for communication with the WebServer process, only for application nodes. Port must be accessible to all other search and application nodes.  <b>If not specified, a dynamic port will be chosen and all ports must be open among the nodes.</b>	(none)	optional	(see description)	(none)
sonar. search. host	Listening IP, only for search nodes. IP must be accessible to all other search and application nodes.	127.0.0.1	<b>mandatory</b> for search nodes	(none)	(see description)
sonar. search. port	Listening port, only for search nodes. Port must be accessible to all other search and application nodes.	9001	optional	(none)	(see description)
sonar. search. initialStateTimeout	The timeout for the Elasticsearch nodes to elect a master node. The default value will be fine in most cases, but in a situation where startup is failing because of a timeout, this may need to be adjusted. The value must be set in the format : {integer}{time unit}. Valid {timeunit} values are : <ul style="list-style-type: none"> <li>• ms : milliseconds</li> <li>• s : seconds</li> <li>• m : minutes</li> <li>• h : hours</li> <li>• d : days</li> <li>• w : weeks</li> </ul>	cluster: 120s  standalone: 30s	optional	(none)	(see description)
sonar. auth. jwtBase64HS256Secret	Required for authentication with multiple web servers. It is used to keep user sessions opened when they are redirected from one web server to another by the load balancer.  See <i>sonar.properties</i> for details about how to generate this secret key.	(none)	<b>mandatory</b>	(see description)	(none)

## Frequently Asked Questions

- Does Elasticsearch discover automatically other ES nodes? **No**. Multicast is disabled. All hosts (IP+port) must be listed.
- Can different nodes run on the same machine? **Yes** but the best is to have 5 machines to be really resilient to failures
- Do the members of a cluster can be discovered automatically? **No**, all nodes must be configured in sonar.properties

## Limitations

- A downtime of the cluster has to be accepted when performing SonarQube upgrade or plugins installations
- There is no way to perform actions on the cluster from a central app - all operations have to be done manually on each node of the cluster
- All application nodes must be stopped when installing, uninstalling or upgrading a plugin
- Plugins are not shared, it means if you install/uninstall/upgrade a given plugin in one application node, you need to do the same actions on the other application node