

# Nagios 101: The Fundamentals

## What do we use Nagios for?

Nagios uses plugins (small, self-contained programs) to check the status of a **host** or **service** (keep reading for an explanation of these two).

Nagios helps IT professionals tell whether certain aspects of a **host** are healthy or not.



### Meet your host:

A **host** is anything that is connected to a network or has an IP address (like, but not limited to, a server, router, printer, etc.).

## How does Nagios work?

### Plugins and Commands

A Nagios user creates a **command** that gives orders to a **plugin** (a small, self-contained program that contains all the instructions for monitoring something) on how it should perform a check for a **host** or a **service**.



### Services

In Nagios, **services** are like recipes that define how to monitor anything from:

- What **plugin** to use
- The target (what you're monitoring)
- Frequency of the check
- What command to use
- Etc.



### Checks

Checks are the actual act of the **plugin** following the service's orders at a specific frequency and threshold to see if something is (for example) up or down.



### Nagios

Nagios processes the results of the checks, puts them into information a human can read, and takes action accordingly (e.g. sending an email notification, restarting a server, etc.).



## How about an example?\*



Let's say you have a fancy coffee maker. That coffee maker (the one with the IP address) is your **host**.

You love coffee and you want to make sure this fancy machine is working well and doing its job. So you monitor it.



First you need a **plugin**, which is a small program that you can write yourself, or download from the Internet. This **plugin** contains all of the information you need to monitor different aspects of your coffee maker.

Like:

- Is it on? -Is there water in the tank?
- Is there coffee in the cup? -Is it the right temperature?

```
<i write my own code />
```

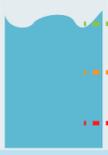
Now that you've got your **plugin**, you need to define how often the plugin should be run against your coffee maker to check the water level, etc. (*psst.* this action of running a **plugin** is called a **check**.)

This definition of how often the check should be run and how many checks should be run is defined in a **service**.

So, let's say your service tells the **plugin** to check the water level every 5 minutes. And then another service tells that same **plugin** (or a different one) to check the temperature every 2 minutes.

We run a check to determine the status of the water level or temperature (or whatever else we want to monitor) on our host coffee maker.

The statuses could look like this:



- OK! (i.e. The water level is full and ready to brew!)
- Warning! (i.e. The water is halfway full, so we're fine but things could get dicey.)
- Critical! (i.e. The water level is too low!)

Nagios takes the status information from the checks and puts it into a human-readable format, so that you see all the information comfortably from your desk. Nagios lets you work on other things with the assurance that your coffee maker is running smoothly, and lets you know when problems arise.

