

Q1. What is Nagios?

You can answer this question by first mentioning that:

Nagios is one of the monitoring tools that is used for Continuous monitoring of systems, applications, services, and business processes etc. in a DevOps culture. In the event of a failure, Nagios can alert technical staff of the problem, allowing them to begin remediation processes before outages affects business processes, end-users, or customers. With Nagios you don't have to explain why an unseen infrastructure outage affect your organization's bottom line.

Few of its important features are:

Nagios Features	
Feature	Description
Monitoring	Its powerful script APIs allow easy monitoring of in-house and custom applications, services, and systems
Visibility & Awareness	It provides a centralized view of the entire monitored IT infrastructure with detailed status information
Problem Remediation	Alert acknowledgments in Nagios, provide communication on known issues and problem response
Proactive Planning	Trending and capacity planning add-ons are there in Nagios to aware you about the aging infrastructure
Reporting	Availability reports ensure SLAs are being met & provide a record of alerts, notifications, and alert response
Customizable Code	Since it is an open source software you get the full access to its source code
Large Community	Nagios is backed up by a community of more than 1 million+ users worldwide which provides free support

Now, once you have defined what is Nagios, you can mention the various things that you can achieve using Nagios.

By using Nagios you can:

- Plan for infrastructure upgrades before outdated systems cause failures.
- Respond to issues at the first sign of a problem.
- Automatically fix problems when they are detected.
- Coordinate technical team responses.

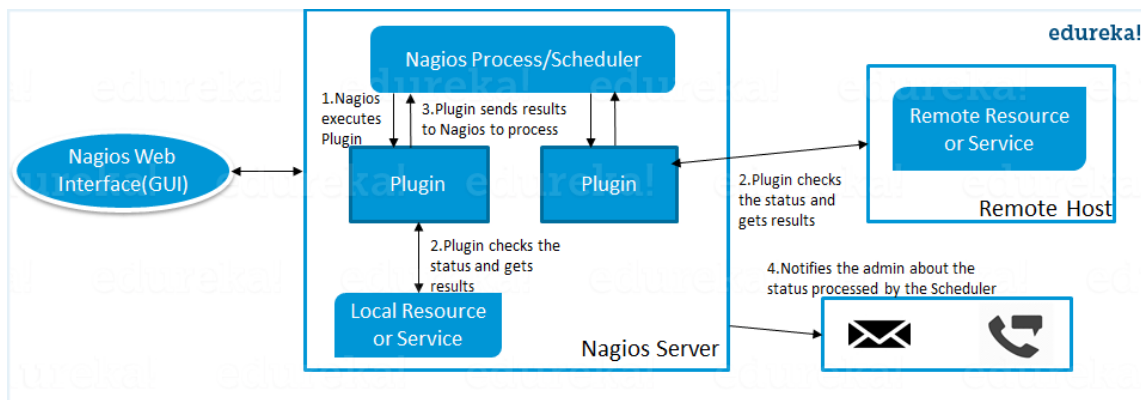
- Ensure your organization's SLAs are being met.
- Ensure IT infrastructure outages have a minimal effect on your organization's bottom line.
- Monitor your entire infrastructure and business processes.

This overall completes the answer to this question. The further details like advantages etc. can be added as per the direction where the discussion is heading.

Q2. How does Nagios work?

I will advise you to follow the below explanation for this answer:

- Nagios runs on a server, usually as a daemon or service.
 - Nagios periodically runs plugins residing on the same server, they contact hosts or servers on your network or on the internet.
 - One can view the status information using the web interface.
 - You can also receive email or SMS notifications if something happens.
 - The Nagios daemon behaves like a scheduler that runs certain scripts at certain moments.
 - It stores the results of those scripts and will run other scripts if these results change.
- Refer the diagram below:



Now, the next set of Nagios interview questions will focus on Nagios components like Plugins, NRPE, etc.

Q3. What are Plugins in Nagios?

Begin this answer by defining Plugins.

Plugins are scripts (Perl scripts, Shell scripts, etc.) that can run from a command line to check the status of a host or service. Nagios uses the results from the plugins to determine the current status of hosts and services on your network.

Once you have defined Plugins I will suggest you to explain why we need plugins.

Nagios will execute a Plugin whenever there is a need to check the status of a host or service. The plugin will perform the check and then simply returns the result to Nagios.

Nagios will process the results that it receives from the Plugin and take the necessary actions.

Q4. What is NRPE (Nagios Remote Plugin Executor) in Nagios?

For this answer first give a small definition of NRPE.

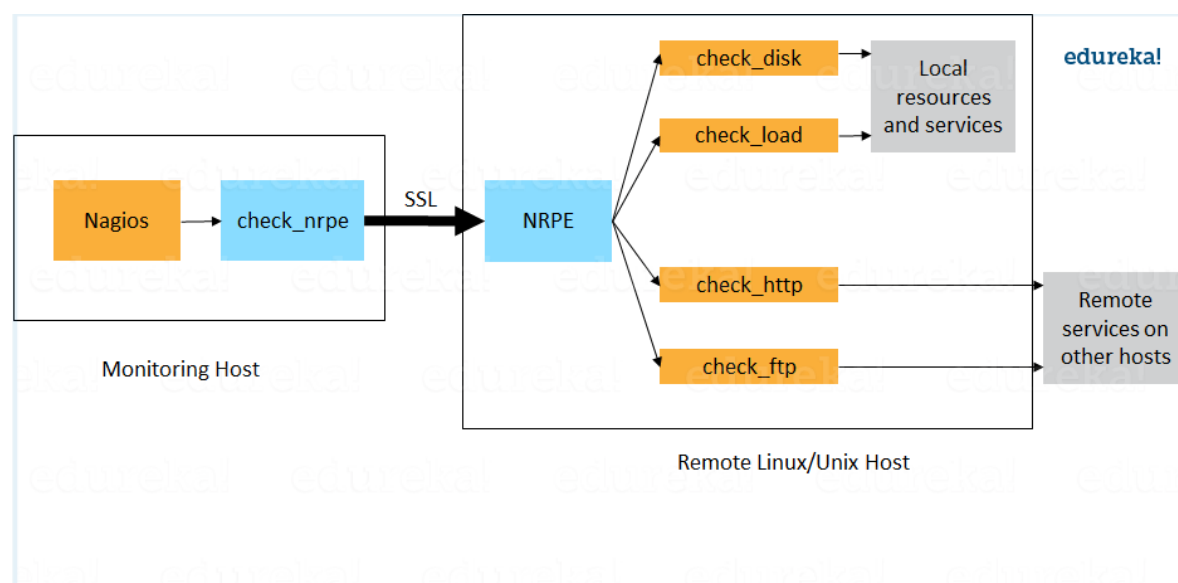
The NRPE add-on is designed to allow you to execute Nagios plugins on remote Linux/Unix machines. The main reason for doing this is to allow Nagios to monitor “local” resources (like CPU load, memory usage, etc.) on remote machines. Since these public resources are not usually exposed to external machines, an agent like NRPE must be installed on the remote Linux/Unix machines.

Now I will advise you to explain the NRPE architecture on the basis of diagram shown below.

The NRPE add-on consists of two pieces:

- The check_nrpe plugin, which resides on the local monitoring machine.
- The NRPE daemon, which runs on the remote Linux/Unix machine.

There is a SSL (Secure Socket Layer) connection between monitoring host and the remote host as shown in the diagram.



Q5. What is meant by Nagios backend?(unable to find a relevant explanation)

My advise will be to follow the below mention flow for this answer:

Both Configuration and Logs can be stored in a backend. Configurations are stored in backend using NagiosQL. Historical data are stored using ndoutils. In addition, you also have nagdb and opdb.

Now, the next set of Nagios interview questions will dig in deep so be prepared.

Q6. What do you mean by passive check in Nagios?

According to me the answer should start by explaining what is Passive check.

Passive checks are initiated and performed by external applications/processes and the Passive check results are submitted to Nagios for processing.

Now I will advise you to explain the need for Passive check.

Passive checks are useful for monitoring services that are Asynchronous in nature and cannot be monitored effectively by polling their status on a regularly scheduled basis. It can also be used for monitoring services that are Located behind a firewall and cannot be checked actively from the monitoring host.

Interviewer will now dig deep, so the next set of Nagios interview questions will test your experience with Nagios.

Q7. When Does Nagios Check for external commands?

Make sure that you stick to the question during your explanation so I will advise you to follow the below mentioned flow:

Nagios check for external commands under the following conditions:

- At regular intervals specified by the `command_check_interval` option in the main configuration file or,
- Immediately after event handlers are executed. This is in addition to the regular cycle of external command checks and is done to provide immediate action if an event handler submits commands to Nagios.

Q8. What is the difference between Active and Passive check in Nagios?

For this answer first point out the basic difference Active and Passive check.

The major difference between Active and Passive checks is that Active checks are initiated and performed by Nagios, while passive checks are performed by external applications.

If your interviewer is looking unconvinced with the above explanation then I will suggest you to also mention some key features of both Active and Passive checks:

Passive checks are useful for monitoring services that are:

- Asynchronous in nature and cannot be monitored effectively by polling their status on a regularly scheduled basis.
- Located behind a firewall and cannot be checked actively from the monitoring host.

The main features of Active checks are as follows:

- Active checks are initiated by the Nagios process.
- Active checks are run on a regularly scheduled basis.

Q9. How does Nagios help with Distributed Monitoring?

Interviewer is expecting an answer related to the distributed architecture of Nagios so I will suggest you to answer it in the below mentioned format:

With Nagios you can monitor your whole enterprise by using a distributed monitoring scheme in which local slave instances of Nagios perform monitoring tasks and report the results back to a single master. You manage all configuration, notification, and reporting from the master, while the slaves do all the work. This design takes advantage of Nagios's ability to utilize passive checks i.e. external applications or processes that send results back to Nagios. In a distributed configuration, these external applications are other instances of Nagios.

Q10. Explain Main Configuration file of Nagios and its location?

I will suggest you to first mention what this main configuration file contains and its function.

The main configuration file contains a number of directives that affect how the Nagios daemon operates. This config file is read by both the Nagios daemon and the CGIs (It specifies the location of your main configuration file).

Now you can tell where it is present and how it is created.

A sample main configuration file is created in the base directory of the Nagios distribution when you run the configure script. The default name of the main configuration file is nagios.cfg, it is usually placed in the etc/ subdirectory of your Nagios installation (i.e. /usr/local/nagios/etc/).

Q11. Explain how Flap Detection works in Nagios?

I will advise you to first explain Flapping first.

Flapping occurs when a service or host changes state too frequently, this causes lot of problem and recovery notifications.

Once you have defined Flapping explain how Nagios detects Flapping.

Whenever Nagios checks the status of a host or service, it will check to see if it has started or stopped flapping. Nagios follow the below procedure to do that:

- Storing the results of the last 21 checks of the host or service analyzing the historical check results and determine where state changes/transitions occur.
- Using the state transitions to determine a percent state change value (a measure of change) for the host or service.
- Comparing the percent state change value against low and high flapping thresholds
- A host or service is determined to have started flapping when its percent state change first exceeds a high flapping threshold.
- A host or service is determined to have stopped flapping when its percent state goes below a low flapping threshold.

Q12. What are the three main variables that affect recursion and inheritance in Nagios?

According to me the proper format for this answer should be:

First name the variables and then a small explanation of each of these variables:

- Name
- Use
- Register

Now I will give a small explanation for each of these variables.

Name is a placeholder that is used by other objects. Use defines the “parent” object whose properties should be used. Register can have a value of 0 (indicating its only a template) and 1 (an actual object). The register value is never inherited.

Q13. What is meant by saying Nagios is Object Oriented?

Answer to this question is pretty direct I will answer this by saying:

One of the features of Nagios is object configuration format in that you can create object definitions that inherit properties from other object definitions and hence the name. This simplifies and clarifies relationships between various components.

Q14. What is State Stalking in Nagios?

I will advise you to first give a small introduction on State Stalking.

State Stalking is used for logging purposes. When Stalking is enabled for a particular host or service, Nagios will watch that host or service very carefully and log any changes it sees in the output of check results. Depending on the discussion between you and interviewer you can also add:

It can be very helpful in later analysis of the log files. Under normal circumstances, the result of a host or service check is only logged if the host or service has changed state since it was last checked.

Q15. Nagios says my machine is unreachable, not down. What is the difference and how it is achieved?

First I will suggest you to explain:

When Nagios says a node is unreachable, a node is unreachable if Nagios is not able to find a path to the node.

Now you can mention the difference.

The node itself may be up but because Nagios is unable to connect to it, it has to mark this as unreachable. To achieve this, Nagios use parent-child relationship between components.

Finally for better understanding explain it with an example.

A router may be defined as a parent for a server.

- Now Nagios checks for server and marks it as down.
- It then checks the parent (in our example, the router)
- If parent is also down, then server is marked as unreachable.
- If Parent is up, the server is marked as really down.

Q16. Explain Nagios state types?

According to me the best way to put this answer is by saying:

The current state of monitored services and hosts is determined by two components:

- The status of service or host i.e. OK, WARNING, UP, DOWN etc..
- The type of state the service or host is in.

There are two types of states SOFT states and HARD states.

Now explain what is Soft and Hard states:

- When a service or host check results are in a non-OK or non-UP state and the service check has not yet been rechecked the number of times specified by the `max_check_attempts` directives in the service or host definition. This is called Soft Error. When a service or a host recovers from Soft Error that is considered as Soft Recovery.
- When a service or host check results are in a non-OK or non-UP state and the service check has been rechecked the number of times specified by the `max_check_attempts` directives in the service or host definition. This is called Hard Error. When a service or a host recovers from Hard Error that is considered as Hard Recovery.

This is the end of my blog on Nagios interview questions and if you want in-depth knowledge about the whole DevOps life-cycle click on the button below: