

5.2 Nagios configuration

1. Overview

PrivateServer appliances can be monitored remotely.

Currently, a set of [Nagios](#) commands is provided to check the reachability, health and resource usage of PrivateServer appliances.

2. Nagios commands for PrivateServer appliances

Remotely monitoring PrivateServer appliances is best done with the PrivateServer -specific Nagios commands. This section describes how to add these commands to your monitoring host, how to enable them on the appliance and the meaning of their outputs.

2.1. Configuring the monitoring host for PrivateServer monitoring

On the monitoring host, the configuration of Nagios has to be modified. This is a simple process in a few steps:

1. define PrivateServer -specific commands
2. Listing PrivateServer appliances on your network
3. verify the configuration
4. restart Nagios

2.1.1. Defining PrivateServer -specific commands

Download [privateserver.commands.cfg](#) and save it to your monitoring host as:

- `/etc/nagios/objects/privateserver.commands.cfg`
- `/usr/local/nagios/etc/objects/privateserver.commands.cfg` (if Nagios was installed from source code)



The file can be saved anywhere, but the suggested locations will put it next to other Nagios configuration files.

Include `privateserver.commands.cfg` from `nagios.cfg` with a line like the following:

```
# replace with the full path to your privateserver.commands.cfg
cfg_file=/etc/nagios/objects/privateserver.commands.cfg
```

2.1.2. Listing PrivateServer appliances on your network

Download [privateserver.host.cfg.template](#). For each of your appliances:

1. make a copy of the template as `appliance hostname.cfg`, under the `/etc/nagios/objects` directory (or `/usr/local/nagios/etc/objects`, if Nagios was installed from source code)
2. open `appliance hostname.cfg` in an editor and:
 - a. replace all occurrences of `localhost.localdomain` with the hostname of the appliance
 - b. replace all occurrences of `127.0.0.1` with the IP address (recommended) or the hostname of the appliance
 - c. save
3. include `appliance hostname.cfg` from another configuration file (e.g. `/etc/nagios/nagios.cfg`) with a line like the following:

```
# replace with the full path to the configuration file
cfg_file=/etc/nagios/objects/host1.privateserver.test.cfg
```

Configuration files created from the template require the following definitions to be already present in your Nagios configuration:

- `linux-server` host definition
- `generic-service` service definition

All of the above definitions are present in the default Nagios configuration, but they may be absent in your installation of Nagios

2.1.3. Verifying the Nagios configuration


Run the following command:

```
nagios -v /etc/nagios/nagios.cfg
```

```
# if you installed Nagios from source code, you might need to run this instead:
/usr/local/nagios/bin/nagios -v /etc/nagios.cfg
```

Review the output, and correct any errors reported by Nagios. **Nagios only reports the first error it finds**, so you will need to verify the configuration after every change, until it reports no errors.

2.1.4. Restarting Nagios

 Always [verify the configuration](#) before starting or restarting Nagios

Run the following command:

```
systemctl restart nagios
```

Nagios will restart with the new configuration. The changes should be immediately visible in the web interface.

2.2. Configuring the appliance for Nagios monitoring

For remote monitoring to work, the appliance must be reachable from the monitoring host. Additionally, most commands need a specific appliance service to be assigned to the network interface used for management (for each command, it will be documented whether this is the case, and which service affects it). The service assignment UI can be found in the web console, under [Server Configuration](#) Applications.

Edit `/etc/nagios/nrpe.cfg` and set `allowed_host` to the IP address of Nagios monitoring server.

Start `nrpe` service:

```
systemctl start nrpe
```

```
systemctl enable nrpe
```


2.3. Reachability checks

The commands in this category check for the reachability of the appliance's administration interfaces.

2.3.1. `check_privateserver_ping`

Checks whether the PrivateServer appliance responds to ping requests. For more information, see the documentation for the Nagios [check_ping](#) plugin.

 This command sends 5 ping requests.

 This command can fail with 100% packet loss if ICMP pings are blocked between the monitoring host and the appliance.

2.3.1.1. Status

Status	Meaning
<div>OK</div>	The appliance is alive
<div>WARNING</div>	Average RTT is larger than 3 seconds, or packet loss is 80% or more
<div>CRITICAL</div>	Average RTT is larger than 5 seconds, or packet loss is 100%

2.3.1.2. Output

A typical, healthy output is similar to:

```
PING OK - Packet loss = 0%, RTA = 33.22 ms
```

2.4. Service health checks

The commands in this category check whether the appliance's services, internal or external, are up and working correctly.

2.4.1. `check_privateserver_sip`

Performs a test call on the PrivateServer appliance to ensure that the SIP service can handle calls correctly. Requires the `nrpe` service to be [enabled on the management network interface](#).

2.4.1.1. Status

Status	Meaning
OK	The SIP service is up and running normally, and can currently handle calls correctly.
WARNING	Both participants to the test call completed the call successfully, but the call was hung up immediately. This can mean the SIP service is responding too slowly, or that the appliance is low on resources. See the output for more information.
CRITICAL	One or both participants to the test call encountered an error. See the output for more information.

2.4.1.2. Output

If the status is **CRITICAL**, the output contains the exit code of both participants to the test call. At least one will be non-zero, indicating an error. ...

2.4.2. check_privateserver_web_console

Checks that the web-based administration interface of the PrivateServer appliance is reachable and running correctly. Requires the `http` service to be [enabled on the management network interface](#).

2.4.2.1. Status

Status	Meaning
OK	The web console is reachable and appears to be running correctly.
WARNING	<ul style="list-style-type: none">Web console reported a client error (HTTP status in the 400 range)HTTPS certificate is about to expire
CRITICAL	<ul style="list-style-type: none">Fatal error connecting to the web consoleI/O error during the requestSyntax error in the responseWeb console reported a server error (HTTP status in the 500 range)Invalid or expired HTTPS certificate

2.4.3. check_privateserver_ssh_console

Checks that the PrivateServer appliance is reachable through SSH. Requires the `ssh` service to be [enabled on the management network interface](#).

2.4.3.1. Status

Status	Meaning
OK	The SSH server is reachable and appears to be running correctly.
WARNING	Should never happen.
CRITICAL	Fatal error connecting to the SSH server, or malformed response from the SSH server.

2.4.4. check_privateserver_db_status

Checks that the database service on the PrivateServer appliance is running correctly. Requires the `nrpe` service to be [enabled on the management network interface](#).

2.4.4.1. Status

Status	Meaning
OK	The database is up and running correctly.
WARNING	Non-fatal error connecting to the server, or no server status available.
CRITICAL	Fatal error connecting to the server, or error querying server status.

2.4.5. check_privateserver_db_data

Checks that the database service on the PrivateServer appliance is responding to queries. Requires the **nrpe** service to be [enabled on the management network interface](#).

2.4.5.1. Status

Status	Meaning
OK	The database is up and running correctly and responding to simple queries.
WARNING	Non-fatal error connecting to the server.
CRITICAL	Fatal error connecting to the server, or error executing the query.

2.5. Resource usage checks

The commands in this category monitor the usage of the appliance's finite resources (CPU, memory, etc.).

2.5.1. check_privateserver_cpu

Checks the CPU usage on the PrivateServer appliance. Requires the **nrpe** service to be [enabled on the management network interface](#).

2.5.1.1. Status

Status	Meaning
OK	CPU usage normal.
WARNING	CPU usage between 90% and 95%.
CRITICAL	CPU usage 95% or above.

2.5.2. check_privateserver_memory

Checks the user and swap memory usage on the PrivateServer appliance. User memory is calculated as total memory usage minus buffers and cache. Requires the **nrpe** service to be [enabled on the management network interface](#).

2.5.2.1. Status

Status	Meaning
OK	Memory usage normal.
WARNING	User memory or swap usage between 90% and 95%.
CRITICAL	User memory or swap usage above 95%.

2.5.3. check_privateserver_disk

Checks the disk space usage on the PrivateServer appliance. Requires the **nrpe** service to be [enabled on the management network interface](#).

2.5.3.1. Status

Status	Meaning
OK	Free disk space normal.
WARNING	Free disk space is 5% or less on any filesystem.
CRITICAL	Free disk space is 0% on any filesystem.

2.5.4. check_privateserver_bandwidth

Checks the network bandwidth usage on the PrivateServer appliance. Requires the `nrpe` service to be [enabled on the management network interface](#).

2.5.4.1. Status

Status	Meaning
OK	Network bandwidth usage normal.
WARNING	Network bandwidth usage between 20 Mb/s and 100 Mb/s on any network interface.
CRITICAL	Network bandwidth usage above 100 Mb/s on any network interface.

3. Appendix: Attachments

File	Modified
File <code>privateserver.commands.cfg</code>	Nov 06, 2020 by Luca Guerrieri
File <code>privateserver.host.cfg.template</code>	Nov 06, 2020 by Luca Guerrieri

[Download All](#)

[5.1 Services and Resources](#)