



Generic SNMP

Contenu du pack

Modèles

Le connecteur de supervision **Generic SNMP** apporte 2 modèles d'hôte :

- **App-Protocol-SNMP-custom**
- **App-Protocol-SNMP-Only-custom**

App-Protocol-SNMP

App-Protocol-SNMP-Only

Non rattachés à un modèle d'hôte

Ce modèle d'hôte n'apporte pas de modèle de service.

Règles de découverte

Découverte d'hôtes

Host

Service

Nom de la règle	Description
SNMP Agents	Découvre les hôtes en interrogeant leur agent SNMP
SNMP v3 Agents	Découvre les hôtes en interrogeant leur agent SNMP en utilisant SNMP version 3
SNMP IP Addresses (RFC 4293)	Découvre les adresses IP en interrogeant un agent SNMP (RFC 4293)

Rendez-vous sur la [documentation dédiée](#) pour en savoir plus sur la découverte automatique d'hôtes.

Découverte de service



Métriques & statuts collectés

Voici le tableau des services pour ce connecteur, détaillant les métriques rattachées à chaque service.

Numeric-Value String-Value Uptime

Pour ce mode spécifique, les noms des métriques dépendront de la configuration effectuée par l'utilisateur.

Prérequis

Configuration SNMP

L'agent SNMP doit être activé et configuré sur l'équipement. Veuillez vous référer à la documentation officielle du constructeur/éditeur. Il se peut que votre équipement nécessite qu'une liste d'adresses autorisées à l'interroger soit paramétrée. Veillez à ce que les adresses des collecteurs Centreon y figurent bien.

Flux réseau

La communication doit être possible sur le port UDP 161 depuis le collecteur Centreon vers le serveur supervisé.

Installer le connecteur de supervision

Pack

La procédure d'installation des connecteurs de supervision diffère légèrement [suivant que votre licence est offline ou online](#).

1. Si la plateforme est configurée avec une licence *online*, l'installation d'un paquet n'est pas requise pour voir apparaître le connecteur dans le menu **Configuration > Connecteurs > Connecteurs de supervision**. Au contraire, si la plateforme utilise une licence *offline*, installez le paquet sur le **serveur central** via la commande correspondant au gestionnaire de paquets associé à sa distribution :



[Alma / RHEL / Oracle Linux 8](#)[Alma / RHEL / Oracle Linux 9](#)[Debian 11 & 12](#)[C](#)

```
apt install centreon-pack-applications-protocol-snmp
```

2. Quel que soit le type de la licence (*online* ou *offline*), installez le connecteur **Generic SNMP** depuis l'interface web et le menu **Configuration > Connecteurs > Connecteurs de supervision**.

Plugin

À partir de Centreon 22.04, il est possible de demander le déploiement automatique du plugin lors de l'utilisation d'un connecteur. Si cette fonctionnalité est activée, et que vous ne souhaitez pas découvrir des éléments pour la première fois, alors cette étape n'est pas requise.

Plus d'informations dans la section [Installer le plugin](#).

Utilisez les commandes ci-dessous en fonction du gestionnaire de paquets de votre système d'exploitation :

[Alma / RHEL / Oracle Linux 8](#)[Alma / RHEL / Oracle Linux 9](#)[Debian 11 & 12](#)[C](#)

```
apt install centreon-plugin-applications-protocol-snmp
```

Utiliser le connecteur de supervision

Utiliser un modèle d'hôte issu du connecteur

[App-Protocol-SNMP-custom](#)[App-Protocol-SNMP-Only-custom](#)

1. Ajoutez un hôte à Centreon depuis la page **Configuration > Hôtes**.
2. Complétez les champs **Nom**, **Alias & IP Address/DNS** correspondant à votre ressource.
3. Appliquez le modèle d'hôte **App-Protocol-SNMP-custom**.

Si vous utilisez SNMP en version 3, vous devez configurer les paramètres spécifiques

associés via la macro **SNMPEXTRAOPTIONS**. Plus d'informations dans la section [Troubleshooting SNMP](#).

Utiliser un modèle de service issu du connecteur

1. Si vous avez utilisé un modèle d'hôte et coché la case **Créer aussi les services liés aux modèles**, les services associés au modèle ont été créés automatiquement, avec les modèles de services correspondants. Sinon, [créez les services désirés manuellement](#) et appliquez-leur un modèle de service.
2. Renseignez les macros désirées (par exemple, ajustez les seuils d'alerte). Les macros indiquées ci-dessous comme requises (**Obligatoire**) doivent être renseignées.

Numeric-Value

String-Value

Uptime

Macro	Description	Valeur par défaut	Obligatoire
FORMAT	Output format (Default: 'current value is %s')	current value is %s	
OID	OID value to check (numeric format only)		
WARNING	Warning threshold		
CRITICAL	Critical threshold		
EXTRAOPTIONS	Any extra option you may want to add to the command (E.g. a --verbose flag). Toutes les options sont listées ici		

3. [Déployez la configuration](#). Le service apparaît dans la liste des services supervisés, et dans la page **Statut des ressources**. La commande envoyée par le connecteur est indiquée dans le panneau de détails du service : celle-ci montre les valeurs des macros.

Comment puis-je tester le plugin et que signifient les options des commandes ?

Une fois le plugin installé, vous pouvez tester celui-ci directement en ligne de commande depuis votre collecteur Centreon en vous connectant avec l'utilisateur **centreon-engine** (`su - centreon-engine`). Vous pouvez tester que le connecteur arrive bien à superviser une ressource en utilisant une commande telle que celle-ci (remplacez les valeurs d'exemple par les vôtres) :

```
/usr/lib/centreon/plugins//centreon_generic_snmp.pl \  
  --plugin=apps::protocols::snmp::plugin \  
  --mode=uptime \  
  --hostname=10.0.0.1 \  
  --snmp-community='my-snmp-community' \  
  --snmp-version=2c \  
  \
```

La commande devrait retourner un message de sortie similaire à :

```
OK: System uptime is: 3m 25s | 'uptime'=205s;;;0;
```

Diagnostic des erreurs communes

Rendez-vous sur la [documentation dédiée](#) pour le diagnostic des erreurs communes des plugins Centreon.

Modes disponibles

Dans la plupart des cas, un mode correspond à un modèle de service. Le mode est renseigné dans la commande d'exécution du connecteur. Dans l'interface de Centreon, il n'est pas nécessaire de les spécifier explicitement, leur utilisation est implicite dès lors que vous utilisez un modèle de service. En revanche, vous devrez spécifier le mode correspondant à ce modèle si vous voulez tester la commande d'exécution du connecteur dans votre terminal.

Tous les modes disponibles peuvent être affichés en ajoutant le paramètre `--list-mode` à la commande :

```
/usr/lib/centreon/plugins//centreon_generic_snmp.pl \  
  --plugin=apps::protocols::snmp::plugin \  
  --list-mode
```

Le plugin apporte les modes suivants :



Mode	Modèle de service associé
cache	Non utilisé dans ce connecteur de supervision
collection	Non utilisé dans ce connecteur de supervision
discovery	Utilisé pour la découverte d'hôtes
dynamic-command	Non utilisé dans ce connecteur de supervision
numeric-value	App-Protocol-SNMP-Numeric-Value-custom
response-time	Non utilisé dans ce connecteur de supervision
string-value	App-Protocol-SNMP-String-Value-custom
uptime	App-Protocol-SNMP-Uptime-custom

Options disponibles

Options génériques

Les options génériques sont listées ci-dessous :

Option	Description
<code>--mode</code>	Define the mode in which you want the plugin to be executed (see <code>--list-mode</code>).
<code>--dyn-mode</code>	Specify a mode with the module's path (advanced).
<code>--list-mode</code>	List all available modes.
<code>--mode-version</code>	Check minimal version of mode. If not, unknown error.
<code>--version</code>	Return the version of the plugin.
<code>--pass-manager</code>	Define the password manager you want to use. Supported managers are: environment, file, keepass, hashicorpvault and teampass.

Option	Description
<code>--verbose</code>	Display extended status information (long output).
<code>--debug</code>	Display debug messages.
<code>--filter-perfdata</code>	Filter perfdata that match the regexp. Eg: adding <code>--filter-perfdata='avg'</code> will remove all metrics that do not contain 'avg' from performance data.
<code>--filter-perfdata-adv</code>	Filter perfdata based on a "if" condition using the following variables: label, value, unit, warning, critical, min, max. Variables must be written either <code>%{variable}</code> or <code>%(variable)</code> . Eg: adding <code>--filter-perfdata-adv='not %(value) == 0 and %(max) eq ""'</code> will remove all metrics whose value equals 0 and that don't have a maximum value.
<code>--explode-perfdata-max</code>	Create a new metric for each metric that comes with a maximum limit. The new metric will be named identically with a <code>'_max'</code> suffix). Eg: it will split <code>'used_prct'=26.93%;0:80;0:90;0;100</code> into <code>'used_prct'=26.93%;0:80;0:90;0;100 'used_prct_max'=100%;;;;</code>
<code>--change-perfdata</code> <code>--extend-perfdata</code>	Change or extend perfdata. Syntax: <code>--extend-perfdata=searchlabel,newlabel,target[, [newuom], [min], [max]]</code> Common examples: Convert storage free perfdata into used: <code>--change-perfdata=free,used,invert()</code> Convert storage free perfdata into used: <code>--change-perfdata=used,free,invert()</code> Scale traffic values automatically: <code>--change-perfdata=traffic,,scale(auto)</code> Scale traffic values in Mbps: <code>--change-perfdata=traffic_in,,scale(Mbps),mbps</code> Change traffic values in percent: <code>--change-perfdata=traffic_in,,percent()</code>
<code>--extend-perfdata-group</code>	Add new aggregated metrics (min, max, average or sum) for groups of metrics defined by a regex match on the metrics' names. Syntax: <code>--extend-perfdata-group=regex,namesofnewmetrics,calculation[, [newuom], [min], [max]]</code> regex: regular expression namesofnewmetrics: how the new metrics' names are composed (can use \$1, \$2... for groups defined by () in regex). calculation: how the values of the new metrics should be calculated newuom (optional): unit of measure ' the new metrics min (optional): lowest value the metrics can reach

Option	Description
	<p>max (optional): highest value the metrics can reach Common examples: Sum wrong packets from all interfaces (with interface need --units-errors=absolute): --extend-perfdata-group=',packets_wrong,sum(packets_(discard error)_(in out))' Sum traffic by interface: --extend-perfdata-group='traffic_in_(.*)',traffic_\$1,sum(traffic_(in out)_\$1)'</p>
--change-short-output --change-long-output	<p>Modify the short/long output that is returned by the plugin. Syntax: --change-short-output=patternreplacementmodifier Most commonly used modifiers are i (case insensitive) and g (replace all occurrences). Eg: adding --change-short-output='OKUpgi' will replace all occurrences of 'OK', 'ok', 'Ok' or 'oK' with 'Up'</p>
--change-exit	<p>Replace an exit code with one of your choice. Eg: adding --change-exit=unknown=critical will result in a CRITICAL state instead of an UNKNOWN state.</p>
--range-perfdata	<p>Rewrite the ranges displayed in the perfdata. Accepted values: 0: nothing is changed. 1: if the lower value of the range is equal to 0, it is removed. 2: remove the thresholds from the perfdata.</p>
--filter-uom	<p>Mask the units when they don't match the given regular expression.</p>
--opt-exit	<p>Replace the exit code in case of an execution error (i.e. wrong option provided, SSH connection refused, timeout, etc). Default: unknown.</p>
--output-ignore-perfdata	<p>Remove all the metrics from the service. The service will still have a status and an output.</p>
--output-ignore-label	<p>Remove the status label ("OK:", "WARNING:", "UNKNOWN:", "CRITICAL:") from the beginning of the output. Eg: 'OK: Ram Total:...' will become 'Ram Total:...'</p>
--output-xml	<p>Return the output in XML format (to send to an XML API).</p>
--output-json	<p>Return the output in JSON format (to send to a JSON API).</p>

Option	Description
<code>--output-openmetrics</code>	Return the output in OpenMetrics format (to send to a tool expecting this format).
<code>--output-file</code>	Write output in file (can be combined with json, xml and openmetrics options). E.g.: <code>--output-file=/tmp/output.txt</code> will write the output in <code>/tmp/output.txt</code> .
<code>--disco-format</code>	Applies only to modes beginning with 'list-'. Returns the list of available macros to configure a service discovery rule (formatted in XML).
<code>--disco-show</code>	Applies only to modes beginning with 'list-'. Returns the list of discovered objects (formatted in XML) for service discovery.
<code>--float-precision</code>	Define the float precision for thresholds (default: 8).
<code>--source-encoding</code>	Define the character encoding of the response sent by the monitored resource Default: 'UTF-8'.
<code>--hostname</code>	Name or address of the host to monitor (mandatory).
<code>--snmp-community</code>	SNMP community (default value: public). It is recommended to use a read-only community.
<code>--snmp-version</code>	Version of the SNMP protocol. 1 for SNMP v1 (default), 2 for SNMP v2c, 3 for SNMP v3.
<code>--snmp-port</code>	UDP port to send the SNMP request to (default: 161).
<code>--snmp-timeout</code>	Time to wait before sending the request again if no reply has been received, in seconds (default: 1). See also <code>--snmp-retries</code> .
<code>--snmp-retries</code>	Maximum number of retries (default: 5).
<code>--maxrepetitions</code>	Max repetitions value (default: 50) (only for SNMP v2 and v3).
<code>--subsetleaf</code>	How many OID values per SNMP request (default: 50) (for <code>get_leaf</code> method. Be cautious when you set it. Prefer to let the default value

Option	Description
<code>--snmp-autoreduce</code>	Progressively reduce the number of requested OIDs in bulk mode. Use it in case of SNMP errors (By default, the number is divided by 2).
<code>--snmp-force-getnext</code>	Use SNMP getNext function in SNMP v2c and v3. This will request one OID at a time.
<code>--snmp-cache-file</code>	Use SNMP cache file.
<code>--snmp-username</code>	SNMP v3 only: User name (securityName).
<code>--authpassphrase</code>	SNMP v3 only: Pass phrase hashed using the authentication protocol defined in the <code>--authprotocol</code> option.
<code>--authprotocol</code>	SNMP v3 only: Authentication protocol: MD5 SHA. Since net-snmp 5.9.1: SHA224 SHA256 SHA384 SHA512.
<code>--privpassphrase</code>	SNMP v3 only: Privacy pass phrase (privPassword) to encrypt messages using the protocol defined in the <code>--privprotocol</code> option.
<code>--privprotocol</code>	SNMP v3 only: Privacy protocol (privProtocol) used to encrypt messages. Supported protocols are: DES AES and since net-snmp 5.9.1: AES192 AES192C AES256 AES256C.
<code>--contextname</code>	SNMP v3 only: Context name (contextName), if relevant for the monitored host.
<code>--contextengineid</code>	SNMP v3 only: Context engine ID (contextEngineID), if relevant for the monitored host, given as a hexadecimal string.
<code>--securityengineid</code>	SNMP v3 only: Security engine ID, given as a hexadecimal string.
<code>--snmp-errors-exit</code>	Expected status in case of SNMP error or timeout. Possible values are warning, critical and unknown (default).

Option	Description
<code>--snmp-tls-transport</code>	Transport protocol for TLS communication (can be: 'dtlsudp', 'tlstcp').
<code>--snmp-tls-our-identity</code>	X.509 certificate to identify ourselves. Can be the path to the certificate file or its contents.
<code>--snmp-tls-their-identity</code>	X.509 certificate to identify the remote host. Can be the path to the certificate file or its contents. This option is unnecessary if the certificate is already trusted by your system.
<code>--snmp-tls-their-hostname</code>	Common Name (CN) expected in the certificate sent by the host if it differs from the value of the <code>--hostname</code> parameter.
<code>--snmp-tls-trust-cert</code>	A trusted CA certificate used to verify a remote host's certificate. If you use this option, you must also define <code>--snmp-tls-their-hostname</code> .

Options des modes

Les options disponibles pour chaque modèle de services sont listées ci-dessous :

Numeric-Value String-Value Uptime

Option	Description
<code>--memcached</code>	Memcached server to use (only one server).
<code>--redis-server</code>	Redis server to use (only one server). Syntax: address[:port]
<code>--redis-attribute</code>	Set Redis Options (<code>--redis-attribute="cnx_timeout=5"</code>).
<code>--redis-db</code>	Set Redis database index.
<code>--failback-file</code>	Failback on a local file if redis connection failed.
<code>--memexpiration</code>	Time to keep data in seconds (Default: 86400).

Option	Description
<code>--statefile-dir</code>	Define the cache directory (default: <code>'/var/lib/centreon/centplugins'</code>).
<code>--statefile-suffix</code>	Define a suffix to customize the statefile name (Default: <code>''</code>).
<code>--statefile-concat-cwd</code>	If used with the <code>'--statefile-dir'</code> option, the latter's value will be used as a sub-directory of the current working directory. Useful on Windows when the plugin is compiled, as the file system and permissions are different from Linux.
<code>--statefile-format</code>	Define the format used to store the cache. Available formats: <code>'dumper'</code> , <code>'storable'</code> , <code>'json'</code> (default).
<code>--statefile-key</code>	Define the key to encrypt/decrypt the cache.
<code>--statefile-cipher</code>	Define the cipher algorithm to encrypt the cache (Default: <code>'AES'</code>).
<code>--oid</code>	OID value to check (numeric format only).
<code>--warning</code>	Warning threshold.
<code>--critical</code>	Critical threshold.
<code>--oid-type</code>	Type of the OID (Default: <code>'gauge'</code>). Can be <code>'counter'</code> also. <code>'counter'</code> will use a retention file.
<code>--counter-per-seconds</code>	Convert counter value on a value per seconds (only with type <code>'counter'</code>).
<code>--extracted-pattern</code>	Set pattern to extracted a number.
<code>--format</code>	Output format (Default: <code>'current value is %s'</code>)
<code>--format-custom</code>	Apply a custom change on the value (Example to multiply the value: <code>--format-custom='* 8'</code>).

Option	Description
<code>--format-scale</code>	Scale bytes value. We'll display value in output.
<code>--format-scale-type</code>	Could be 'network' (value divide by 1000) or 'other' (divide by 1024) (Default: 'other') Output format (Default: 'current value is %s')
<code>--perfddata-unit</code>	Perfddata unit in perfddata output (Default: '')
<code>--perfddata-name</code>	Perfddata name in perfddata output (Default: 'value')
<code>--perfddata-min</code>	Minimum value to add in perfddata output (Default: '')
<code>--perfddata-max</code>	Maximum value to add in perfddata output (Default: '')
<code>--config-json</code>	JSON format to configure the mode. Can check multiple OID. Example: <code>--config-json='[{ "oid": ".1.3.6.1.2.1.1.3.0", "perfddata_name": "oid1", "format": "current oid1 value is %s"}, { "oid": ".1.3.6.1.2.1.1.3.2", "perfddata_name": "oid2", "format": "current oid2 value is %s"}]'</code>

Pour un mode, la liste de toutes les options disponibles et leur signification peut être affichée en ajoutant le paramètre `--help` à la commande :

```
/usr/lib/centreon/plugins//centreon_generic_snmp.pl \  
  --plugin=apps::protocols::snmp::plugin \  
  --mode=numeric-value \  
  --help
```

Dernière mise à jour le **15 avr. 2025**

[Éditer cette page](#)

[Envoyer du feedback sur cette page](#)

[Questionner la communauté](#)

