

# Documentation du système de supervision

- Serveur nagios 192.168.0.60
- Sonde numéro 1:
- Nom de la sonde: charge CPU
- Objectif de la sonde: vérifier le niveau d'utilisation du CPU
- Paramétrage de la sonde (valeur pour warning et critical ):

```
define command {
```

```
    command_name      check_local_load
    command_line       $USER1$/check_load -w $ARG1$ -c $ARG2$
}
```

```
define service {
```

```
    use                local-service
    host_name           nagios-server
    service_description charge CPU
    check_command       check_local_load!.70,.60,.50!.90,.80,.70!/
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- vérifier les processus en cause via HTOP
- les arrêter si possible

- Serveur nagios 192.168.0.60
- Sonde numéro 2:
- Nom de la sonde: espace disque
- Objectif de la sonde: verifier l'espace disque restant
- Paramétrage de la sonde (valeur pour warning et critical ):

```
define command {
```

```
    command_name    check_local_disk
    command_line     $USER1$/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$
}
```

```
define service {
```

```
    use                local-service
    host_name           nagios-server
    service_description espace libre
    check_command       check_local_disk!30%!20%!/
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- Nettoyer le disque de ses fichiers inutiles
- Ajouter des disques supplémentaires si besoin

- Serveur nagios 192.168.0.60
- Sonde numéro 3:
- Nom de la sonde: usage ram
- Objectif de la sonde: verifier le taux de ram disponible
- Paramétrage de la sonde (valeur pour warning et critical):

```
define command {
    command_name    check_local_mem
    command_line    $USER1$/check_mem.pl -w $ARG1$ -c $ARG2$
}
```

```
define service {
    use                local-service
    host_name          nagios-server
    service_description usage RAM
    check_command       check_local_mem!70!85!
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- vérifier les processus qui consomme le plus de RAM avec HTOP et les stopper si possible
- Augmenter la capacité de RAM disponible

- Serveur nagios 192.168.0.60
- Sonde numéro 4:
- Nom de la sonde: curent users
- Objectif de la sonde: verifie qu'une seule session est ouverte sur le serveur
- Paramétrage de la sonde (valeur pour warning et critical):

```
define command {
```

```
    command_name    check_local_users
    command_line     $USER1$/check_users -w $ARG1$ -c $ARG2$
}
```

```
define service {
```

```
    use                local-service
    host_name           nagios-server
    service_description Current Users
    check_command        check_local_users!1!1!
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- vérifier dans les logs qui est connecté ou s'est connecté

lnav /var/log/auth.log

- Serveur nagios 192.168.0.60
- Sonde numéro 5:
- Nom de la sonde: HTTP
- Objectif de la sonde: vérifier le fonctionnement du service HTTP
- Paramétrage de la sonde (valeur pour warning et critical):

```
define command {
```

```
    command_name    check_http
    command_line     $USER1$/check_http -I $HOSTADDRESS$ $ARG1$
}
```

```
define service {
```

```
    use                local-service
    host_name           nagios-server
    service_description HTTP
    check_command        check_http
    notifications_enabled 0
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- vérifier le statut du service apache via `systemctl status apache2.service`
- redémarrer le service via `systemctl restart apache2.service`
- vérifier les logs en cas d'erreurs

serveur wordpress

- Serveur wordpress 192.168.0.61
- Sonde numéro 1:
- Nom de la sonde: charge CPU
- Objectif de la sonde: vérifier le niveau d'utilisation du CPU
- Paramétrage de la sonde (valeur pour warning et critical):

#### sur le serveur wordpress :

`command[check_local_load]=/usr/lib/nagios/plugins/check_load -r -w .70,.60,.50 -c .90,.80,.70`

#### sur le serveur nagios :

```
define service {
    use                generic-service
    host_name          serv-web
    service_description charge CPU
    check_command       check_nrpe!check_local_load!
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- vérifier les processus en cause via HTOP
- les arrêter si possible

- Serveur wordpress 192.168.0.61
- Sonde numéro 2:
- Nom de la sonde: espace disque
- Objectif de la sonde: verifier l'espace disque restant
- Paramétrage de la sonde (valeur pour warning et critical):

#### **sur le serveur wordpress :**

`command[check_local_disk]=/usr/lib/nagios/plugins/check_disk -w 30% -c 20% -p/`

#### **sur le serveur nagios :**

```
define service {
    use                generic-service
    host_name          serv-web
    service_description espace libre
    check_command       check_nrpe!check_local_disk!
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- Nettoyer le disque de ses fichiers inutiles
- Ajouter des disques supplémentaires si besoin



- Serveur wordpress 192.168.0.61
- Sonde numéro 3:
- Nom de la sonde: charge RAM
- Objectif de la sonde: verifier le taux de ram disponible
- Paramétrage de la sonde (valeur pour warning et critical ):

#### sur le serveur wordpress :

`command[check_local_mem]=/usr/lib/nagios/plugins/check_mem.pl -w 70 -c 85`

#### sur le serveur nagios :

```
define service {
    use                generic-service
    host_name          serv-web
    service_description usage RAM
    check_command       check_nrpe!check_local_mem!
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- vérifier les processus qui consomme le plus de RAM avec HTOP et les stopper si possible
- Augmenter la capacité de RAM disponible

- Serveur wordpress 192.168.0.61
- Sonde numéro 4:
- Nom de la sonde: processus apache2
- Objectif de la sonde: verifie que les processus apaches tournent
- Paramétrage de la sonde (valeur pour warning et critical ):

#### sur le serveur wordpress :

`command[check_proc_apache]=/usr/lib/nagios/plugins/check_procs -C apache2 -c 1:`

#### sur le serveur nagios :

```
define service {
    use                generic-service
    host_name          serv-web
    service_description processus apache actif
    check_command       check_nrpe!check_proc_apache
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- vérifier le statut du service apache via `systemctl status apache2.service`
- redémarrer le service via `systemctl restart apache2.service`
- vérifier les logs en cas d'erreurs

- Serveur wordpress 192.168.0.61
- Sonde numéro 5:
- Nom de la sonde: page d'index
- Objectif de la sonde: verifie que la page d'index est bien presente
- Paramétrage de la sonde (valeur pour warning et critical ):

#### sur le serveur wordpress :

`command[check_index]=usr/lib/nagios/plugins/check_http -I 192.168.0.61 -u /wordpress/index.php`

#### sur le serveur nagios :

```
define service {
    use                generic-service
    host_name          serv-web
    service_description index
    check_command       check_nrpe!check_index
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- verifier le statut du service apache via `systemctl status apache2.service`
- redémarrer le service via `systemctl restart apache2.service`
- vérifier les logs en cas d'erreurs
- vérifier les fichiers du site wordpress

- Serveur wordpress 192.168.0.61
- Sonde numéro 6:
- Nom de la sonde: etat mysql
- Objectif de la sonde: vérifie si le service mysql tourne
- Paramétrage de la sonde (valeur pour warning et critical):

#### sur le serveur wordpress :

`command[check_mysql]=/usr/lib/nagios/plugins/check_mysql -u wordpress -p wp123`

#### sur le serveur nagios :

```
define service {
    use                generic-service
    host_name          serv-web
    service_description mysql
    check_command       check_nrpe!check_mysql
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- verifier le statut du service mysql via `systemctl status mysql.service`
- redémarrer le service via `systemctl restart mysql.service`
- vérifier les logs en cas d'erreurs

- Serveur wordpress 192.168.0.61
- Sonde numéro 7:
- Nom de la sonde: nombre users
- Objectif de la sonde: vérifie qu'une seule session est ouverte sur le serveur
- Paramétrage de la sonde (valeur pour warning et critical):

#### sur le serveur wordpress :

`command[check_local_users]=/usr/lib/nagios/plugins/check_users -w 1 -c 1`

#### sur le serveur nagios :

```
define service {
    use                generic-service
    host_name          serv-web
    service_description current users
    check_command       check_nrpe!check_local_users!
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- vérifier dans les logs qui est connecté ou s'est connecté

`lnav /var/log/auth.log`

- Serveur wordpress 192.168.0.61
- Sonde numéro 8:
- Nom de la sonde: HTTP
- Objectif de la sonde: vérifier le fonctionnement du service HTTP
- Paramétrage de la sonde (valeur pour warning et critical): /

### sur le serveur wordpress :

`command[check_http]=/usr/lib/nagios/plugins/check_http -I 192.168.0.61`

### sur le serveur nagios :

```
define service {
    use                generic-service
    host_name          serv-web
    service_description acces http
    check_command       check_nrpe!check_http!
}
```

Action à entreprendre en cas d'anomalie (dont les commandes Linux correspondantes):

- vérifier le statut du service apache via `systemctl status apache2.service`
- redémarrer le service via `systemctl restart apache2.service`
- vérifier les logs en cas d'erreurs