

# Stats CPU by WinRM

## Sommaire

- Contexte
- Paramétrage
  - Données utilisées provenant des modèles
    - Données communes pour les checks des modèles
    - Données spécifiques pour ce check
    - Données DFE ( Duplicate Foreach )
  - Données utilisées provenant du check
  - Données globales
  - Propriétés de l'hôte
- Résultat
  - Exemple
  - Interprétation
    - Statut
    - Résultat
    - Résultat Long
- Métriques
  - Définition
  - Exemple
- Erreurs et pré-requis
  - Erreurs de connexion ( communes à tous les checks )
    - UNKNOWN – Transport error : failed to send request: request timed out
    - UNKNOWN – Transport error : sent request failed: connection refused
    - UNKNOWN – Transport error : sent request failed: host is not reachable
    - UNKNOWN – Transport error : sent request failed: DNS resolution failed
    - UNKNOWN – Transport error : failed to build request: given uri is invalid
    - UNKNOWN – Authentication NTLM failed : NTLM is not supported by the server
    - UNKNOWN – Authentication NTLM failed : Unauthorized
    - UNKNOWN – Authentication Basic failed : Basic is not supported by the server
    - UNKNOWN – Authentication Basic failed : Unauthorized
  - Erreurs de configuration de l'hôte à superviser ( communes à tous les checks )
    - UNKNOWN – Response fault error: Code: s:Sender, Subcode: w:AccessDenied, Reason: Access is denied.
    - MONITORED HOST - BAD STATE – Command execution Failed. Permission denied.
    - UNKNOWN – Command execution Failed. [...] Provider failure

## Objectifs



### Pré-requis

Le module **livedata-module-sla-provider** doit être activé sur le **broker-module-livedata** pour que la route **/api/v1/sla/** soit accessible.

La configuration du module se trouve par défaut dans le fichier suivant : **/etc/shinken/modules/livedata-module-sla-provider.cfg** : [Le livedata-module-sla-provider](#)

Méthode POST de type READ qui permet de recevoir la liste des données SLA de tous les éléments ( *hôtes, clusters et checks* ) :

- Filtrées ( *optionnel* ),
- Rangées,
  - En arbres ( *hôtes/clusters checks* ),
  - Tous au même niveau,
- En choisissant,
  - les informations présentes dans le retour de la requête ( *optionnel* ),
  - la période sur laquelle les données SLA seront récupérées ( *optionnel* ),
    - les données SLA sont calculées à la fin de la journée, donc la dernière donnée disponible est celle de la veille,
  - le nombre d'éléments par page ( *optionnel* ).
- Les données SLA récupérées sont triées dans l'ordre chronologie ( du plus récent au plus vieux ).

## Paramètres

Pour définir l'appel, 5 paramètres sont disponibles :

- standards :
  - **filterX**
  - **checks\_in\_tree**
  - **ouput\_field**
  - **period**
  - **page\_settings**

## filterX ( Filtres )

Les filtres ont pour formes :

- **filterX** : **expression-expression**
  - ~ ayant le sens de "et"
  - **expression** de la forme : **critère:valeur0<sup>^</sup>valeur1**
    - où <sup>^</sup> a le sens de "ou"
- X vaut de 0 à 9.
- Chaque élément correspondant à **au moins un des filtres** sera retourné.

Les **critères** suivants sont utilisables :

- type
- father\_name
- father\_uuid
- father\_templates
- check\_name
- check\_uuid
- address
- realm
- host\_groups
- notification\_contacts
- notification\_contact\_groups
- business\_impact

Vous pouvez trouver la description de ces filtres dans la page [V1 - Les paramètres de l'API du module livedata-module-sla-provider du broker-module-livedata](#)

## checks\_in\_tree ( Rangées sous forme d'arbre )

Ce paramètre permet d'indiquer où sont situés les checks dans la réponse de retour :

- **true** : les checks sont accrochés à leurs hôtes / clusters ( *forme d'arbre* )
- **false** : les checks sont listés au niveau des hôtes / clusters ( *une liste* )

REMARQUE : Dans le cas où le filtre vaut uniquement **type=check** ( *donc pas d'hôtes ou clusters* )

- Si le **checks\_in\_tree = true**, les hôtes / clusters seront quand même présents pour les checks correspondant à ce filtre.
- Si le **checks\_in\_tree = false**, les hôtes et clusters ne sont pas présents.

## output\_field ( Informations présentes dans le retour de la requête )

Ce paramètre permet de lister les champs qui seront affichés sur le résultat en sortie.

- Les champs présents par défaut sont :
  - type
  - father\_name
  - father\_uuid
  - check\_name
  - check\_uuid

Les champs ci-dessus sont décrits dans la page [V1 - Les paramètres de l'API du module livedata-module-sla-provider du broker-module-livedata](#)

- Les champs suivants sont propres à cette route et sont présent par défaut :

Nom	Format	Description
sla_total	Secondes	Temps total de SLA ( 86400 secondes étant 1 journée complète )
sla_missing	Secondes	Temps en statut <b>Données manquantes</b>
sla_ok	Secondes	Temps en statut <b>OK</b>
sla_inactive	Secondes	Temps en statut <b>Shinken Inactive</b>
sla_unknown	Secondes	Temps en statut <b>Inconnu</b>
sla_crit	Secondes	Temps en statut <b>Critique</b>
sla_warn	Secondes	Temps en statut <b>Attention</b>

<b>sla_thresholds</b>	Liste de pourcentages	Deux pourcentages : <ul style="list-style-type: none"> <li>la première valeur est le seuil d'avertissement</li> <li>la deuxième valeur est le seuil de critique</li> </ul> Les pourcentages ont une précision à 3 chiffres ( <i>ex: 90.001</i> )
<b>sla_date</b>	Chaîne de caractères	au format <b>jj_mm_aaaa</b> ( <i>ex: 12_05_2021</i> )

### period ( entre quelles dates de début et de fin, prendre les données SLA )

Nom	Valeur par défaut	Description et syntaxe
<b>period=start:date~end:date</b>	La dernière heure	Définit la période où collecter les données SLA <ul style="list-style-type: none"> <li>Les dates sont au format <b>jj_mm_aaaa</b> ( <i>ex: 12_05_2021</i> )</li> <li>Si <b>start</b> n'est <b>pas précisé</b>, cela signifie que le début de la période est la date actuelle.</li> <li>Si <b>end</b> n'est <b>pas précisé</b>, la fin de la période est maintenant.</li> </ul>

### page\_settings ( combien d'éléments par page et quelle page retourner )

L'API peut, grâce à ce paramètre, définir le nombre d'éléments par page et le numéro de la page retournée, ce qui permet de contrôler le volume d'échange de données. Ceci est possible vu que les données SLA sont figées en base de données.

Le champ **has\_next\_page** dans la partie **pagination** du retour permet de savoir s'il y a une page suivante.

Nom	Valeur par défaut	Info
<b>page_settings=nb_element:size</b>	Le nombre d'éléments par défaut d'une page est <b>100</b>	<b>Pour la première requête</b> <ul style="list-style-type: none"> <li><b>nb_element</b> étant la taille de la page</li> </ul>
<b>page_settings=page:page_index~nb_element:size</b>	Le nombre d'éléments par défaut d'une page est <b>100</b>	<ul style="list-style-type: none"> <li><b>nb_element</b> étant la taille de la page</li> <li><b>page</b> étant l'index de page demandée. Les indexes de page commencent à 0</li> </ul>



Si le **check\_in\_tree** est à *True*, le nombre d'éléments par page correspondra aux hôtes / clusters.

Si le **check\_in\_tree** est à *False*, le nombre d'éléments par page correspondra aux hôtes / clusters / checks.

### Exemple

Exemple permettant d'obtenir la première page d'une requête renvoyant 100 éléments avec leurs données SLA, du début de l'année 2021 au 1er mai 2021.

```
curl -s -S -H "x-api-token: XYZ" \
-d "period=01_01_2021:~end:01_05_2021" \
-d "page_settings=page:0~nb_element:100" \
http://broker-module-livedata:50100/api/v1/sla
```

Exemple permettant d'obtenir la quatrième page d'une requête renvoyant 100 éléments avec leurs données SLA, du début de l'année 2021 au 1er mai 2021.

```
curl -s -S -H "x-api-token: XYZ" \
-d "period=01_01_2021:~end:01_05_2021" \
-d "page_settings=page:4~nb_element:100" \
http://broker-module-livedata:50100/api/v1/sla
```

### Réponse

## Codes de retour

Codes de retour	Explications
200	OK
400	Paramètre invalide
401	Accès nécessite une authentification ou un Token valide.
403	Authentification de l'utilisateur OK, mais droits non suffisants.
500	L'appel est valide, mais un problème d'exécution est survenu.

## Retour du code 200

En premier apparaîtra des informations donnant le nombre d'éléments :

- **elements\_count**
  - nb\_elements\_observable
  - nb\_elements\_filtered
  - nb\_elements\_in\_page
  - nb\_elements\_unmatched
  - nb\_fathers\_total\_observable
  - nb\_fathers\_filtered
  - nb\_fathers\_in\_page
  - nb\_fathers\_unmatched
  - nb\_checks\_total\_observable
  - nb\_checks\_filtered
  - nb\_checks\_in\_page
  - nb\_checks\_unmatched

Avec une information supplémentaire propre à la route :

Nom	Format	Description
nb_sla_in_page	Entier	Nombre de données SLA présentes dans la page

En deuxième position. Les données SLA vont être retournées dans le format suivant :

- **pagination**
  - has\_next\_page
  - nb\_total\_page
  - page
  - page\_size

Les champs présents pour chaque élément retourné possédant des données SLA doivent être choisis avec l'option **output\_field**, mais les champs suivants sont au minimum automatiquement retournés :

**Checks\_in\_tree à True :**

- **fathers\_found**
    - type
    - father\_name
    - father\_uuid
    - checks
      - type
      - check\_name
      - check\_uuid
      - sla
        - sla\_total
        - sla\_missing
        - sla\_ok
        - sla\_inactive
        - sla\_unknown
        - sla\_crit
        - sla\_warn
        - sla\_thresholds
        - sla\_date
- **fathers\_not\_found**

**Checks\_in\_tree à False :**

- **elements\_found**
  - type
  - father\_name / check\_name
  - father\_uuid / check\_uuid
  - sla
    - sla\_total
    - sla\_missing
    - sla\_ok
    - sla\_inactive
    - sla\_unknown
    - sla\_crit
    - sla\_warn
    - sla\_thresholds
    - sla\_date
- **elements\_not\_found**

Voir la page [V1 - Les champs présents dans le retour 200 des API du broker-module-livedata](#), pour la description complète de tous les champs pouvant être renvoyés. Pour les éléments non trouvés, les critères utilisés dans les filtres seront renvoyés dans la description des l'élément.

### Exemple 1 : checks\_in\_tree=true

```
curl -s -S -H "x-api-token: XYZ" \  
-d "period=start:10_05_2021~end:11_05_2021" \  
-d "checks_in_tree=true" \  
-d "filter01=type:check" \  
-d "page_settings=p:0~s:1" \  
http://broker-module-livedata:50100/api/v1/sla
```

```

{
  "elements_count": {
    "nb_clusters_filtered": 0,
    "nb_clusters_in_page": 0,
    "nb_hosts_filtered": 2,
    "nb_clusters_observable": 0,
    "nb_elements_observable": 32,
    "nb_checks_filtered": 29,
    "nb_checks_in_page": 1,
    "nb_hosts_observable": 3,
    "nb_checks_observable": 29,
    "nb_hosts_in_page": 1,
    "nb_elements_filtered": 31,
    "nb_elements_in_page": 2,
    "nb_sla_in_page": 2
  },
  "pagination": {
    "nb_total_page": 2,
    "page": 0,
    "page_size": 1,
    "has_next_page": true
  },
  "fathers_found": [{
    "father_name": "Host 01",
    "father_uuid": "f87c2e56b94b11ebaf7e080027c44e8f",
    "type": "host",
    "checks": [{
      "check_name": "Check 01",
      "type": "check_host",
      "check_uuid": "f87c2e56b94b11ebaf7e080027c44e8f-fdd0c038b94b11ebb21f080027c44e8f",
      "sla": [{
        "sla_date": "10_05_2021",
        "sla_total": 86400,
        "sla_warn": 0,
        "sla_unknown": 0,
        "sla_thresholds": [99.0, 97.0],
        "sla_missing": 0,
        "sla_ok": 0,
        "sla_inactive": 86400,
        "sla_crit": 0
      }, {
        "sla_date": "11_05_2021",
        "sla_total": 86400,
        "sla_warn": 0,
        "sla_unknown": 0,
        "sla_thresholds": [99.0, 97.0],
        "sla_missing": 0,
        "sla_ok": 0,
        "sla_inactive": 86400,
        "sla_crit": 0
      }
    ]
  }
]
}

```

## Exemple 2 : checks\_in\_tree=false

```

curl -s -S -H "x-api-token: XYZ" \
-d "period=start:10_05_2021~end:11_05_2021" \
-d "checks_in_tree=false" \
-d "filter01=type:check" \
-d "page_settings=p:0~s:1" \
http://broker-module-livedata:50100/api/v1/sla

```

```

{
  "elements_count": {
    "nb_clusters_filtered": 0,
    "nb_clusters_in_page": 0,
    "nb_hosts_filtered": 0,
    "nb_clusters_observable": 0,
    "nb_elements_observable": 32,
    "nb_checks_filtered": 29,
    "nb_checks_in_page": 0,
    "nb_hosts_observable": 3,
    "nb_checks_observable": 29,
    "nb_hosts_in_page": 0,
    "nb_elements_filtered": 29,
    "nb_elements_in_page": 0,
    "nb_sla_in_page": 2
  },
  "pagination": {
    "nb_total_page": 29,
    "page": 0,
    "page_size": 1,
    "has_next_page": true
  },
  "elements_found": [{
    "check_name": "Check 01",
    "type": "check_host",
    "father_uuid": "f87c2e56b94b11ebaf7e080027c44e8f",
    "father_name": "Host 01",
    "check_uuid": "f87c2e56b94b11ebaf7e080027c44e8f-fdd0c038b94b11ebb21f080027c44e8f",
    "sla": [{
      "sla_date": "10_05_2021",
      "sla_total": 86400,
      "sla_warn": 0,
      "sla_unknown": 0,
      "sla_thresholds": [99.0, 97.0],
      "sla_missing": 0,
      "sla_ok": 0,
      "sla_inactive": 86400,
      "sla_crit": 0
    }, {
      "sla_date": "11_05_2021",
      "sla_total": 86400,
      "sla_warn": 0,
      "sla_unknown": 0,
      "sla_thresholds": [99.0, 97.0],
      "sla_missing": 0,
      "sla_ok": 0,
      "sla_inactive": 86400,
      "sla_crit": 0
    }
  ]
}
]
}
}

```