

Olá colegas, hoje quero falar de uma necessidade que surgiu do nada aqui na empresa, que foi criar a arvore de SLA do Zabbix.

Quando chegou o pedido ele foi dessa forma, precisamos disso para amanhã, então respirei fundo e pensei como posso fazer isso rápido, eu tenho mais de mil itens monitorados na ferramenta.

A solução foi usar o nosso amigo Python e um pouco de criatividade com os códigos rsrs.

Então sem demoras vamos a melhor parte que é fazer isso funcionar.

OBS. Neste código vou postar como remover a arvore antiga e criar uma nova se o seu caso for como o meu em que não existe a arvore de SLA mantenha comentado o trecho do “def” que remove a mesma ok, outro ponto recomendo fortemente que teste e ajuste o fonte se necessário usando alguma ferramenta que não esteja em produção. E deixo claro que qualquer erro, ou problema é exclusivamente de sua responsabilidade.

## **Primeiramente vamos conectar no Zabbix.**

```
import urllib.request as urllib2  
  
from zabbix_api import ZabbixAPI  
  
zapi = ZabbixAPI(server="http://192.168.120.193/zabbix")  
  
zapi.login("Admin", "senha do usuario")
```

Agora criar as def para Grupos, Hosts, Triggers e Itens

```
def obterGrupos():  
  
    hostgroups = zapi.hostgroup.get({ "output": "extend", "real_hosts": True })  
  
    listaGrupos = []  
  
    for x in hostgroups: listaGrupos += [x['name']]
```

```
    return listaGrupos
```

---

```
def obterGrupold(nomeGrupo):
```

```
    groupId = zapi.hostgroup.get({ "output": "extend", "filter": { "name": nomeGrupo } })[0]['groupid']
```

```
    return groupId
```

---

```
def obterHosts(nomeGrupo):
```

```
    hosts_grupo = zapi.host.get({ "groupids": obterGrupold(nomeGrupo), "output": [ "host" ] })
```

```
    listaHosts = []
```

```
    for x in hosts_grupo: listaHosts += [x['host']]
```

```
    return listaHosts
```

---

```
def obterHostId(nomeHost):
```

```
    hostId = zapi.host.get({ "output": "hostid", "filter": { "host": nomeHost } })[0]['hostid']
```

```
    return hostId
```

---

```
def obterTriggersHosts(nomeHost):
```

```
    triggers = zapi.trigger.get({ "hostids": obterHostId(nomeHost), "expandDescription": "true", "expandComment": "true", "expandExpression": "true" })
```

```
    for x in triggers: print (x['description'])
```

---

```
def obterItens(nomeHost):
```

```
    itens = zapi.item.get({ "hostids": obterHostId(nomeHost), "with_triggers": True, "selectTriggers": "extend" })
```

```
listaltens = []

for x in itens: listaltens += [x['name']]

return listaltens
```

---

```
def obterItemTriggerId(nomeHost,nomeItem):

    triggerId = zapi.item.get({ "output": "triggers","hostids":obterHostId(nomeHost), "with_triggers": True, "selectTriggers": "triggerid", "filter": { "name": nomeItem } })[0]['triggers'][0]['triggerid']

    return triggerId
```

E a partir desse ponto vamos começar as def para criar a arvore de SLA

```
def criarServicosPai(nomeGrupo):

    zapi.service.create({ "name": nomeGrupo, "algorithm": "1","showsla": "1",
"goodsla": "99.99", "sortorder": "1" })
```

---

```
def obterServicosPai(nomeGrupo):

    parentId = zapi.service.get({ "selectParent": "extend", "selectTrigger": "extend",
"expandExpression": "true", "filter": { "name": nomeGrupo }})[0]['serviceid']

    return parentId
```

---

```
def criarServicosFilho(nomeHost,nomeGrupo):

    zapi.service.create({ "name": nomeHost, "algorithm": "1", "showsla": "1",
"goodsla": "99.99", "sortorder": "1", "parentid": obterServicosPai(nomeGrupo)})
```

---

```
def obterServicosFilho(nomeHost):

    parentIdChild = zapi.service.get({ "selectParent": "extend", "selectTrigger": "extend",
"expandExpression": "true", "filter": { "name": nomeHost } })[0]['serviceid']
```

```
    return parentIdChild
```

---

```
def criarItensServicos(nomeHost,nomeItem):
```

```
    zapi.service.create({ "name": nomeItem, "algorithm": "1", "showsla": "1",  
"goodsla": "99.99", "sortorder": "1", "parentid": obterServicosFilho(nomeHost),  
"triggerid": obterItemTriggerId(nomeHost,nomeItem)})
```

---

```
def obterServicos():
```

```
    itServices = zapi.service.get({ "selectParent": "extend", "selectTrigger":  
"extend" })
```

```
    listaServicos = []
```

```
    for x in itServices: listaServicos += [x['serviceid']]
```

```
    return listaServicos
```

Neste ponto vou criar um def que remove a arvore de SLA antiga.

```
def removerArvoreServicos():
```

```
    for x in obterServicos():
```

```
        zapi.service.deletedependencies([x])
```

```
        zapi.service.delete([x])
```

Por fim criamos a nossa nova arvore de SLA com nossos itens e triggers

```
def criarArvoreServicos():
```

```
    # removerArvoreServicos()
```

```
    for nomeGrupo in obterGrupos():
```

```
        criarServicosPai(nomeGrupo)
```

```
for nomeHost in obterHosts(nomeGrupo):
```

```
    criarServicosFilho(nomeHost, nomeGrupo)
```

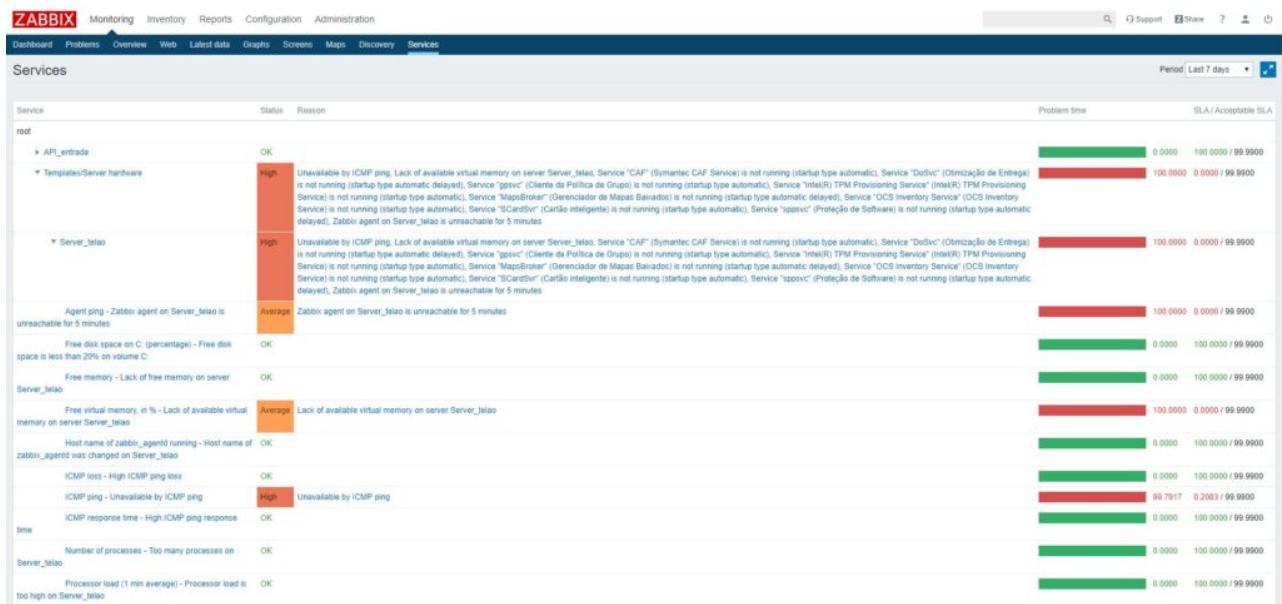
```
    for nomeItem in obterItens(nomeHost):
```

```
        criarItensServicos(nomeHost, nomeItem)
```

```
criarArvoreServicos()
```

Observem que assim como mencionei acima, deixei a função responsável por deletar a arvore antiga comentada. (Tomem cuidado com essa opção)

No fim vocês terão um resultado semelhante a este:



Por hora é isso pessoal, espero que todos tenham gostado e caso queiram dar uma força, curtam lá no linkedin.

Abraços.