Para fazer esse monitoramento, eu utilizei o sistema de monitoramento Zabbix(, sistema Open Source de monitoramento, escalável para grandes ambientes, juntamente com um dos mais conhecidos testadores de internet, o SpeedTest.net By Ookla.

Primeiramente vamos precisar de um servidor com o sistema Zabbix configurado e com acesso a internet para realização do teste. Caso sua empresa tenha diversas unidades e possua o Zabbix Proxy configurado em cada unidade, você pode utilizar o Zabbix Proxy da unidade para realizar o teste de velocidade da internet da mesma.

Você deve acessar o servidor do Zabbix que fará o monitoramento e baixar o Speedtest CLI, ferramenta oficial do Speedtest.net, que fará o teste de velocidade para nós.

No site [**https://www.speedtest.net/apps/cli**](https://www.speedtest.net/apps/cli), podemos encontrar detalhes da ferramenta e utilizar a opção de instalação de acordo com o nosso SO. Utilizarei o exemplo abaixo, demonstrando a instalação em um CentOS.

Texto

Descrição gerada automaticamente

Primeiro executamos o comando abaixo para instalar o WGET, caso ainda não tenham instalado.

# sudo yum install wget

Depois devemos baixar o repositório da Ookla, que possui o Speedtest e em seguida movê-lo para o diretório padrão dos repositórios do CentOS.

# wget https://bintray.com/ookla/rhel/rpm -O bintray-ookla-rhel.repo

# sudo mv bintray-ookla-rhel.repo /etc/yum.repos.d/

Outra forma que pode funcionar

#yum install python3-pip

#pip3 install speedtest-cli

Com o repositório instalado, podemos iniciar a instalação do Speedtest. Basta executar o comando abaixo para instalar.

# sudo yum install speedtest

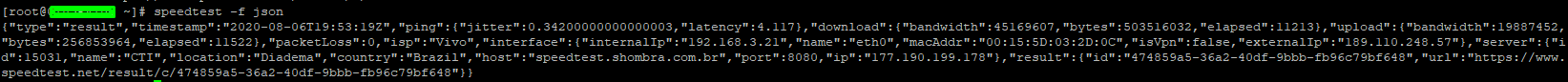
Pronto o SpeedTest está instalado e você pode executa-lo para realizar o seu teste de velocidade.

Digite "speedtest" na linha de comando para iniciar o teste de velocidade. Será exibido dados como na imagem a seguir. Será exibido o servidor onde foi realizado o teste, o ISP(Seu provedor), Latência, Velocidade de Download e Upload, Perda de Pacote e URL com o resultado do teste.

Texto

Descrição gerada automaticamente

Além disso, de acordo com a documentação dessa aplicação do speedtest, é possivel obter o resultado desse teste no formato JSON, igual o exemplo abaixo. Utilizaremos dele para realizar nosso monitoramento.



# speedtest –json \*\*

No Zabbix uma das formas flexíveis de monitoramento, quando não conseguimos monitorar com os recursos nativos da ferramenta, é utilizando da aplicação Zabbix Trapper, que nada mais é que um executável que envia os parâmetros que forem passados para ele, para o Zabbix. Para isso, devemos inicialmente garantir que temos o Zabbix Trapper instalado no servidor. Podemos instala-lo utilizando o comando abaixo:

# yum install zabbix-trapper

# dnf install zabbix-sender

Com o Zabbix Trapper instalado, criei um pequeno script que vai ser executado pelo CRON do Linux de tempos em tempos, para realizar o teste de velocidade e enviar os dados via Zabbix Trapper para o Zabbix.

O Script abaixo deve ser instalado na pasta de ExternalScripts do seu Zabbix. Normalmente localizado no diretório "/usr/lib/zabbix/externalscripts".

Criamos o script "speedtest.sh", com o conteúdo abaixo:

#! /bin/bash

SPEEDTESTRESULT=$(speedtest -f json)

zabbix\_sender -z ZABBIX\_SERVER\_IP -s "SPEEDTEST\_HOSTNAME" -k speedtestX -o "$SPEEDTESTRESULT"

Esse script executará o comando do speedtest no formato JSON, armazenará na variável SPEEDTESTRESULT e depois via zabbix\_sender, enviará as informações via Zabbix Trapper para o Servidor do Zabbix.

Importante ressaltar que o IP do Zabbix Server e o Hostname informado no Script deverá casar com as informações do Host monitorado pelo Zabbix, assim como a chave do item monitorado, no caso "speedtestX".

Com o script criado, devemos dar a permissão de executar para esse script. Portanto podemos rodar o comando abaixo para isso:

# chmod +x /usr/lib/zabbix/externalscripts/speedtest.sh

A parte do Script está pronta, agora devemos programar esse script para ser executado de X em X tempos, para enviar os dados do teste de velocidade para o Zabbix. Para isso, vamos utilizar o CRONTAB do Linux. Digitamos o comando abaixo para abrir o crontab e inserirmos o nosso agendamento.

# crontab -e

Agora inserimos a linha do agendamento. O modelo abaixo fará a coleta do resultado do teste de velocidade às 0h e a cada hora durante todo o dia, e enviará o resultado para o Zabbix.

0 0,\*/1 \* \* \* /usr/lib/zabbix/externalscripts/speedtest.sh

Agora que temos a coleta de dados configurada, devemos garantir que temos um Host que irá receber os dados via Zabbix Trapper. Para ajudar, criei um template que pode ser baixado pelo link abaixo.

[**https://share.zabbix.com/cat-app/internet-speed-test**](https://share.zabbix.com/cat-app/internet-speed-test)

Podemos atribuir o template ao Host que quisermos receber os dados do monitoramento, lembrando de ajustar o script para o nome do Host que você atribuiu o template.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

O funcionamento se dará da seguinte forma:

* O CRON do Linux irá executar o script no tempo programado
* O Script irá invocar o teste de velocidade do Speedtest.net trazendo o resultado em formado JSON
* Esse JSON será enviado via Zabbix Trapper para o Host que identificamos
* O Zabbix receberá o dado no formato JSON no item "X SpeedTest RAW".
* Os itens dependentes farão o tratamento das informações utilizando o Proprocessamento JSON Path
* As informações serão exibidas nos dados recentes do Host.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Com base nisso, podemos criar triggers de acordo com nossa necessidade. No template eu deixei sem as triggers, assim você pode alterar o template para atender a sua necessidade. Com a coleta de dados periódica, podemos ter um gráfico na linha do tempo e utilizar para diversas finalidades, questionar a operadora, criar um dashboard no Grafana, etc.