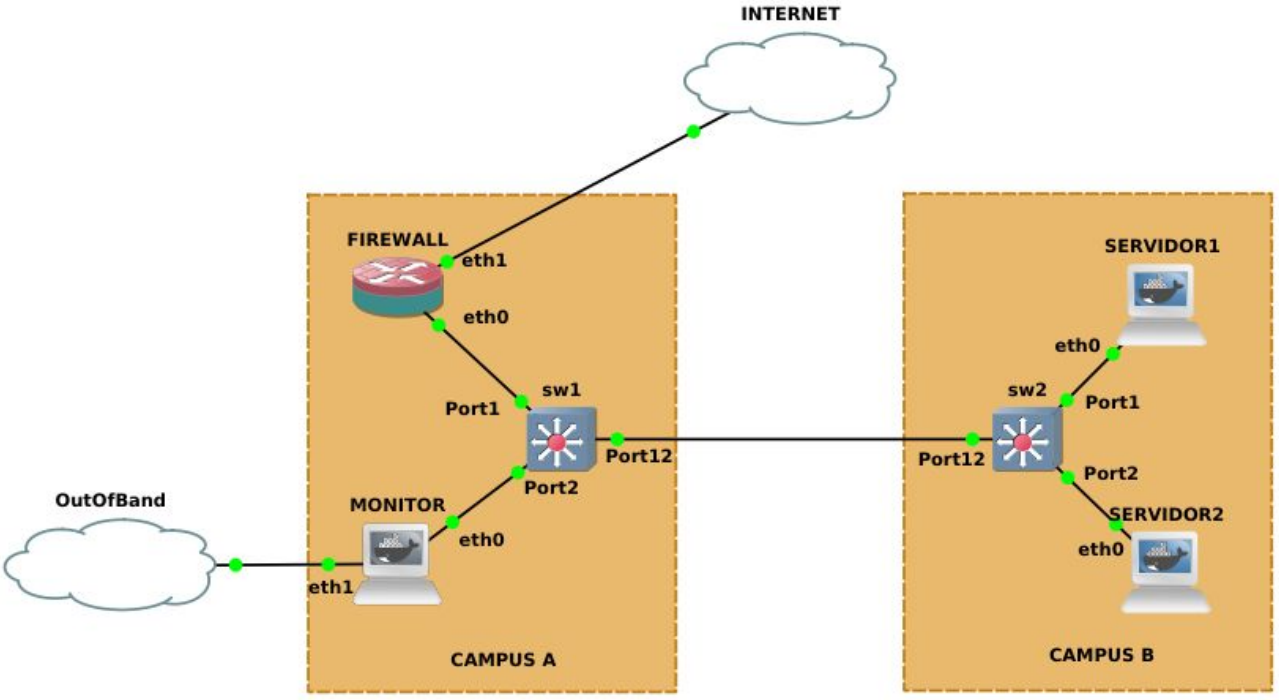
PoP-BA	**XI WTR do POP-BA** XI Workshop de Tecnologias de Redes Ponto de Presença da RNP na Bahia Instrutores: Ibirisol Fontes e Jundaí Abdon	RNP 30 anos

Prática 1

Cenário:



Após realizar a instalação de uma nova máquina (MONITOR, vide figura acima) para fazer monitoramento da infraestrutura de redes e sistema da instituição, para ambos os campi, sua equipe foi incubida de configurar, adequadamente, as ferramentas *Prometheus* e *Grafana*.

Objetivo:

Monitorar os ativos na infraestrutura da organização, é aconselhado que a equipe verifique os requisitos de monitoramento baseado nos serviços e equipamentos da infraestrutura.

Acessando o ambiente:

Acessar o simulador de redes GNS3 e abrir a prática chamada "pratica01.gns3". Clicar no botão de PLAY (verde) no menu e aguardar o carregamento dos ativos. Dois cliques sobre cada ativo para abertura de terminal para execução de comandos.

Ações:

Geral

1. Clicar com o botão direito sobre a máquina *MONITOR*, selecionar "Auxiliary Console", dar enter e executar o comando abaixo:

```
# ip addr show dev eth1
11: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
    link/ether a6:2d:98:43:d6:8b brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.47/24 brd 192.168.122.255 scope global eth1
       valid_lft forever preferred_lft forever
```

No comando acima o **<ip OutOfBand>** é "**192.168.122.47**".

Prometheus

2. Para o primeiro contato com o *Prometheus*, através de um navegador na máquina virtual, acesse a URL abaixo (utilizar o IP Out-Of-Band obtido no item acima).

```
http://<ip OutOfBand>:9090/
```

3. Verifique os menus "**Graph**", "**Status -> Configuration**", "**Status -> Targets**", entre outros menus que gerarem curiosidade.
   a. Entre os *targets* o **snmp**, apenas de exemplo, estará **DOWN.**
4. Em "**Status -> Targets**" é importante observar os seguintes campos.
   a. **Endpoint:** de onde o *Prometheus* faz pull dos dados
   b. **State:** estado do target
   c. **Labels:** parâmetros passados para coleta de dados
   d. **Last Scrape:** tempo em que ocorreu a última coleta
   e. **Scrape Duration:** o tempo de resposta da última coleta
   f. **Error:** saída de erro

SNMP Exporter

5. Agora, é necessário definir quais protocolos e ferramentas serão usados(as) no monitoramento de cada ativo.
   a. SNMP para os switches
   b. Node Exporter para os servidores
6. Configure o suporte ao protocolo SNMP nos switches, para isso, acesse os switches *sw1* e *sw2 (login: admin e senha: <enter>)*:

```
configure snmpv3 add community "secret" name "secret" user "v1v2c_ro"
```

```
enable snmp access snmp-v1v2c
disable snmp access snmpv3
save
```

7. Volte ao terminal auxiliar de *MONITOR* e edite o arquivo "*/etc/prometheus/conf.d/custom/sw1.json*" substituindo o IP *127.0.0.1* pelo IP do *SW1*, conforme abaixo.

```
[
  {
    "targets": [ "192.168.20.201" ],
    "labels": {
      "Hostname": "sw1",
      "Module": "curso_v2"
    }
  }
]
```

8. Verificar na interface do *Prometheus* "**Status -> Targets**" se o status do target snmp está **UP.**
   a. Observe os campos: **Last Scrape, Scrape Duration** e **Error**
9. Agora, através de um navegador na máquina virtual, acesse o *SNMP Exporter* através da URL abaixo.

```
http://<ip OutOfBand>:9116/snmp?module=curso_v2&target=192.168.20.201
```

10. Agora procure pelo valor da variável de tempo de consulta (**snmp_scrape_walk_duration_seconds**)
    a. Verifique se esse valor está mais alto que o **Scrape Duration** configurado no *Prometheus*. Se sim, o que isso implica no estado do *target*? Qual é a relação do intervalo entre coletas e a duração da consulta?
11. Edite o arquivo de configuração do *Prometheus* (**/etc/prometheus/prometheus.yml**) para que o **scrape_interval** do **snmp** seja superior ao tempo de duração da consulta ao switch *SW1*.

```
- job_name: 'snmp'
  scrape_interval: 90s
```

12. É preciso recarregar o Prometheus após a edição da configuração para que seja aplicado a instância em execução.

```
systemctl restart prometheus
```

13. Verifique o funcionamento dos ***targets*** no *Prometheus*.
    a. Qual é a duração da consulta ao *snmp exporter*?
14. Agora crie um arquivo com as configurações de monitoramento do *SW2*, basta acrescentar o arquivo "**/etc/prometheus/conf.d/custom/sw2.json**" com o seguinte conteúdo.

```
[
  {
    "targets": [ "192.168.20.202" ],
    "labels": {
      "Hostname": "sw2",
      "Module": "curso_v2"
    }
  }
]
```

15. Volte ao menu "**Status -> Targets**" do *Prometheus* e verifique se o **status** dos *targets snmp*.
    a. Observe os campos: **Last Scrape, Scrape Duration** e **Error**
16. Verifique novamente no *snmp exporter* o resultado da consulta (PULL) para o *SW1 e SW2*, através das URLs abaixo.

```
http://<ip OutOfBand>:9116/snmp?module=curso_v2&target=192.168.20.201
http://<ip OutOfBand>:9116/snmp?module=curso_v2&target=192.168.20.202
```

17. Procure pelo valor da variável de tempo de consulta (**snmp_scrape_walk_duration_seconds**)
    a. Verifique se esse valor está mais alto que o **Scrape Duration** do *Prometheus*. Se sim, o que isso implica no estado do *target*? É a mesma situação anterior de *SW1*?
18. Edite novamente o arquivo de configuração do *Prometheus* (**/etc/prometheus/prometheus.yml**) e adicione variável abaixo no **snmp** para que seja superior ao tempo de duração da consulta ao switch *SW1*.

```
- job_name: 'snmp'
  scrape_interval: 90s
  scrape_timeout: 60s
```

19. Reinicie o *Prometheus* novamente para que a nova configuração tenha efeito na instância em execução.

```
systemctl restart prometheus
```

20. Verifique o funcionamento dos ***targets*** no *Prometheus*.
    a. Qual é a duração da consulta ao *snmp exporter*?
21. Agora, usaremos uma consulta otimizada do SNMP, apenas com as OIDs necessárias, para isso edite o arquivo "**/etc/prometheus/conf.d/custom/sw2.json**" conforme a seguir (passaremos um parametros para o *snmp exporter* utilizar uma configuração customizada de consulta ao *SW2*).

```
[
  {
    "targets": [ "192.168.20.202" ],
    "labels": {
      "Hostname": "sw2",
      "Module": "curso_v3"
```

```
        }
      }
    ]
```

22. Volte ao menu "**Status -> Targets**" do *Prometheus* e verifique o **Scrape Duration**.

Node Exporter

23. Por fim, é necessário expandir o nosso monitoramento para os outros ativos da rede, levaremos nossa coleta até o *SERVIDOR1* e *SERVIDOR2*.
    a. Extraia e execute o *node exporter*, conforme comandos abaixo, será exportado todas a métricas associadas ao sistema (Não exportaremos as informações de RAID do GNU/Linux, por isto a opção "**--no-collector.mdadm**", pois não existe no ativos emulados).

```
# cd /root && tar -zxvf node_exporter-1.0.0-rc.1.linux-amd64.tar.gz
# /root/node_exporter-1.0.0-rc.1.linux-amd64/node_exporter --no-collector.mdadm
```

24. Agora edite o arquivo de configuração do *Prometheus* (**/etc/prometheus/prometheus.yml**) e adicione as linhas destacadas abaixo, no módulo **node**, para que o *Service Discovery* (através de arquivos *JSON*) possa ser utilizado para descobrir novos nós.

```
 - job_name: 'node'
    scrape_interval: 250ms
    static_configs:
     - targets: ['localhost:9100']
    file_sd_configs:
      - files:
        - '/etc/prometheus/conf.d/node/*.json'
```

25. Crie o diretório que será usado para as novas configurações dos ativos e reinicie o *Prometheus*.

```
mkdir  /etc/prometheus/conf.d/node
systemctl restart prometheus
```

26. *Por fim, basta criar o arquivo de configuração (**/etc/prometheus/conf.d/node/servidores.json**) dos novos nós SERVIDOR1 e SERVIDOR2 (192.168.20.1 e 192.168.20.2 respectivamente), abaixo o conteúdo (inclusive com a porta do node exporter).*

```
[
  {
    "targets": [ "192.168.20.1:9100", "192.168.20.2:9100" ],
    "labels": {
      "job": "servidores"
```

```
    }
  }
]
```

27. Volte ao menu "**Status -> Targets**" do *Prometheus* e verifique se o status dos servidores está **UP.**
    a. Verifique se o tempo de resposta da consulta está de acordo com o "**Scrape Duration**"
    b. Sugira um valor para configuração do "**Scrape Interval**" no *Prometheus*.
    c. Agora edite o arquivo de configuração do *Prometheus* (**/etc/prometheus/prometheus.yml**) e adicione o tempo sugerido para o **scrape_interval**, do módulo **node.**

```
- job_name: 'node'
   scrape_interval: <tempo_sugerido>
   static_configs:
    - targets: ['localhost:9100']
```

    d. Verifique se o valor sugerido foi suficiente para que o "**State**" dos servidores esteja "**UP**". Caso contrário, teste um novo valor. Uma dica para ter uma ideia de valor aproximado, é verificar o tempo de download das métricas para o "MONITOR", é necessário que o tempo entre coletas esteja maior que o tempo para que a consulta termine:

```
wget http://192.168.20.1:9100/metrics -O /dev/null
--2020-09-16 19:00:14--  http://192.168.20.1:9100/metrics
Connecting to 192.168.20.1:9100... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/plain]
Saving to: '/dev/null'

/dev/null           [    <=>             ]  58.08K  76.7KB/s    in 0.8s

2020-09-16 19:00:15 (76.7 KB/s) - '/dev/null' saved [59478]
```

TSDB

28. Agora, vá ao menu "**Graph**" do *Prometheus* para consultarmos o TSDB, verifique os valores retornados para a chave **snmp_scrape_duration_seconds.**
    a. Basta colocar **snmp_scrape_duration_seconds** no campo "**Expression**" e apertar e "**Execute**" (Será retornado o último valor da chave).
29. Amplie o intervalo de consulta das chaves retornadas ao TSDB, basta usar a seguinte consulta abaixo (o retorno é no formato **value@timestamp**).
    a. Quantas amostras foram retornadas?

```
snmp_scrape_duration_seconds[10m]
```

30. Para visualizar um gráfico é necessário trabalhar os valores absolutos ao longo do intervalo temporal selecionado, pois é preciso apenas um valor por ponto plotado, para isso faremos uma média.

```
rate(snmp_scrape_duration_seconds[10m])
```

31. Para visualizar a plotagem, basta clicar na aba "Graph".

Grafana

32. Agora, usaremos uma visualização gráfica mais completa para nossas métricas, para isso acesse o *Grafana* pela URL abaixo.

```
http://<ip OutOfBand>:3000/
```

33. Será solicitado autenticação, basta usar os seguintes dados:
    a. Email or username: **admin**
    b. Password: **admin**
34. Será solicitada a troca da senha (recomendados que seja trocada quando a ferramenta for instanciada na organização **:)**).
35. Navegue pela ferramenta e descubra o que já foi adicionado.
    a. **Create:** Criar e importar dashboards
    b. **Dashboards**
        i. **Home:** visão geral de todos dashboards em uso
        ii. **Manage:** gerenciar os dashboards instalados disponíveis
        iii. **Playlists:** criar visualizações cíclicas de dashboards que rotacionam com o tempo. Úteis para telas de monitoramento.
    c. **Explore:** realizar testes de querys e criação de gráficos
    d. **Alerting:**
        i. **Alert Rules:** visualizar os alertas criados
        ii. **Notification channels:** gerenciar os canais para recebimento de alertas (e mails, chat e etc)
    e. **Configuration:**
        i. **Data Sources:** gerenciar os data sources configurados e adicionar outros
        ii. **Plugins:** gerenciar os plugins instalados
36. O recurso de provisionamento do grafana, permite que o administrador mantenha as configurações da ferramenta fora do banco de dados, e possa especificar a partir de um arquivo de texto, permitindo maior portabilidade das configurações da aplicação. Para verificar como está configurado o **Data Source** e os **Dashboards** vá até o diretório "**/etc/grafana/provisioning**" e verifique o conteúdo dos arquivos.

37. O *Grafana* tem uma grande comunidade, que compartilha diversos templates de dashboards, plugins, etc.
    a. O **SNMP Stats** permite uma visualização interessante para os ativos monitorados via SNMP, disponível na URL abaixo para download.

---

https://grafana.com/grafana/dashboards/11169

---

    b. O **Node Exporter Full** permite uma visualização bem completa dos recursos monitorados através de um *Node Exporter*, disponível na URL abaixo para download.

---

https://grafana.com/grafana/dashboards/1860

---

38. Faça a importação de dois dashboards apresentados acima, basta acessar o menu "**+**" e acessar o item "**Import**".
    a. Coloque o ID do **SNMP Stats** (**11169**) no campo "Import via grafana.com" e aperte em "Load". Basta preencher os campos da próxima tela e colocar o Data Source *Prometheus*, caso tenha identificador duplicado basta mudar manualmente, e importar.
    b. Volte ao menu de importação e coloque o ID do **Node Exporter Full** (**1860**) no campo "Import via grafana.com" e aperte em "Load". Basta preencher os campos da próxima tela e colocar o Data Source *Prometheus*, caso tenha identificador duplicado basta mudar manualmente, e importar.
39. Navegue no menu dos Dashboards e verifique os gráficos importados e as métricas apresentadas.

Alertas

40. Crie um novo *Dashboard* com dois painéis, um com os gráficos da métrica "**snmp_scrape_duration_seconds**" e outro "**node_scrape_collector_duration_seconds**", todos com origem ao *Data Source* do *Prometheus*.
41. Agora, crie um alerta na aba "**Alert**".
    a. Atenção aos campos "**Evaluate every**" e "**For**" (Tempo entre as avaliações da regra de alerta e o tempo de permanência na avaliação que dispara a notificação, respectivamente)
    b. A seção "**Conditions**" também possui as informações relevantes para definir a transgressão do limiar definido. Informe limiares que estejam próximos dos picos apresentados no gráfico de cada painel (Atenção aos tempos de consulta para cada ativo).
    c. Basta, salvar o
42. Basta salvar o painel e acompanhar os limiares serem transgredidos e plotados no gráfico. Também é possível acessar o menu "**Alerting**" (Ícone de sino do painel do lado

esquerdo) para observar as notificações geradas (que não irão para canal de comunicação externo neste cenário).

43. É possível alterar as propriedades do enlace entre os switches *SW1* e *SW2* para observar a variação do tempo de consulta. Basta, no *GNS3*, clicar com o botão esquerdo do mouse e ir na opção "**Packet filters**".
    a. Modifique os parâmetros na aba "**Delay**" e veja o resultado.
44. Por último, **se divirta** com o setup da ferramenta e faça os testes que quiser.

Dicas:

Instalação de programas

Caso precise de algum utilitário nos hosts Linux, basta executar os seguintes comandos:

```
apt update && apt install <nome do pacote ou programa>
```

Por exemplo, para instalar o editor de texto nano, basta:

```
apt update && apt install nano
```

Execução do cenário sem a VM

Para baixar a imagem que usaremos, basta executar:

```
docker pull ibirisol/wtr:debian-monitoring
```

Para executar o container de monitoramento, basta usar o comando abaixo.

```
docker run --rm -ti --cap-add SYS_ADMIN -v /sys/fs/cgroup:/sys/fs/cgroup:ro -p 9090:9090 -p 9100:9100 -p 3000:3000 -p 9116:9116 --name monitoring ibirisol/wtr:debian-monitoring
```

Para editar as configurações no container, basta executar um terminal com o seguinte comando.

```
docker exec -ti monitoring /bin/bash
```

Por fim, para acessar os serviços informados na seção “**Ações**”, basta que no lugar do **<ip OutOfBand>** use “**localhost**”.

Boa atividade!