

# Sua JVM no Prometheus

---

O monitoramento e a visibilidade do ambiente sempre foram de grande importância para avaliação de performance, prevenção de problemas e correção de falhas. Atualmente ferramentas como o Prometheus podem coletar métricas de nodes e serviços para nos auxiliar nessas questões, e o Grafana nos ajuda a ter uma visibilidade melhor dessas métricas com a criação de dashboards e gráficos.

Porém sempre foi um desafio realizar o monitoramento de uma JVM, neste post veremos como realizar a coleta de métricas de uma JVM Wildfly com Prometheus e Grafana utilizando um exporter do Prometheus. As novas versões de Wildfly (a partir da 15), já possuem um subsystem que geram uma página de métricas e que pode ser acessada normalmente pelo browser, assim não se faz necessário a utilização do exporter do Prometheus. Em nosso laboratório iremos demonstrar como realizar a coleta em versões anteriores a 15, pois estas que não possuem o subsystem se tornam um desafio em poder ter uma visibilidade da JVM.

## **Pré-requisitos:**

- Utilizar um SO Linux
- GIT (<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>)
- Docker e Docker Compose (<https://docs.docker.com/install/linux/docker-ce/ubuntu/> | <https://docs.docker.com/compose/install/>)
- Wildfly 14 ou anterior (<https://wildfly.org/downloads/>)

---

## Download do Projeto

Todos os arquivos necessários para a execução do nosso laboratório estão em um repositório do GitHub, assim com o git instalado em sua máquina, escolha um diretório para ser o seu workspace e realize o clone do projeto com o seguinte comando:

```
git clone https://github.com/Caeir0ps/wildfly-exporter.git
```

## Habilitando as Métricas

Após o clone do repositório iremos então começar a configurar nosso Wildfly para exportar as métricas que serão coletadas pelo Prometheus. O exporter que nós iremos utilizar é mantido pela comunidade e pode ser encontrado no repositório "[https://github.com/nlighten/wildfly\\_exporter](https://github.com/nlighten/wildfly_exporter)". Para iniciar a configuração iremos executar os seguintes passos:

- Salve o arquivo 'wildfly\_exporter\_module-0.0.5.jar' que foi baixado do repositório no diretório "\$JBOSS\_HOME/modules/" da sua instância do Wildfly. - A variável \$JBOSS\_HOME indica o diretório do seu Wildfly, para realizar a cópia coloque o caminho relativo ou absoluto no comando da cópia, ex:

```
cp wildfly-exporter/wildfly_exporter_module-0.0.5.jar  
/opt/wildfly/modules/
```

- Acesse o diretório "\$JBASS\_HOME/modules/" e extraia o arquivo JAR com o comando:

```
jar -xvf wildfly_exporter_module-0.0.5.jar
```

Após extrair os arquivos, vamos deletar os seguintes itens que não serão utilizados:

```
rm -rf META-INF ; rm -f wildfly_exporter_module-0.0.5.jar
```

Adicionar os seguintes parâmetros no arquivo de configuração xml utilizado:

```
<global-modules>
  <module name="nl.nlighten.prometheus.wildfly" services="true" meta-
inf="true"/>
</global-modules>
```

```
<subsystem xmlns="urn:jboss:domain:undertow:7.0" default-server="default-
server" default-virtual-host="default-host" default-servlet-
container="default" default-security-domain="other" statistics-
enabled="true">
```

... (Realizar a configuração abaixo para cada Data Source utilizada na instância)

```
<datasource jndi-name="java:jboss/datasources/ExampleDS" pool-
name="ExampleDS" enabled="true"
      use-java-context="true" statistics-enabled="true">
```

Realizar o deploy do WAR em anexo (wildfly\_exporter\_servlet-0.0.5.war)

## Prometheus + Grafana + Docker

Para podermos provisionar nossos servidores Prometheus e Grafana de forma simples e ágil, iremos utilizar contêineres Docker. Partimos do ponto que você já possui Docker e Docker Compose instalados na sua máquina, caso não tenha acesse nosso post que irá te auxiliar na instalação desta ferramenta.

Abaixo está o nosso arquivo docker-compose.file que será utilizado para subir os nossos serviços. As imagens utilizadas são as oficiais

Container Prometheus e Grafana

- Realizar a instalação do Docker-ce e do Docker Compose;
- Salvar os arquivos em anexo "prometheus-compose.zip" e "Wildfly Stats.json";

- Extrair o arquivo "prometheus-compose.zip";
- Editar o arquivo extraído em "./configs/prometheus.yml" adicionando os endereços IPs:Portas das instâncias como no exemplo:
  - job\_name: 'jmx-exporter' metrics\_path: /metrics static\_configs:
    - targets:
      - 172.16.0.198:58080
  - job\_name: 'wildfly-exporter' metrics\_path: /metrics static\_configs:
    - targets:
      - 172.16.0.198:8080
- Executar o comando docker-compose up -d

(O PROMETHEUS USARÁ A PORTA 9090 E O GRAFANA A PORTA 3000, PODE SER ALTERADO NO ARQUIVO DO COMPOSE)