

## Prometheus – Introdução



Prometheus



DevOps  
Mão na  
Massa

# O que é e para que serve?

---

- Ferramenta de monitoria
- Criada para monitorar ambientes dinâmicos (containers) – k8s, swarm, etc.
- Também funciona como monitor bare metal
- Muito popular em infraestrutura de microsserviços e containers



Prometheus

# Por que utilizar?

---

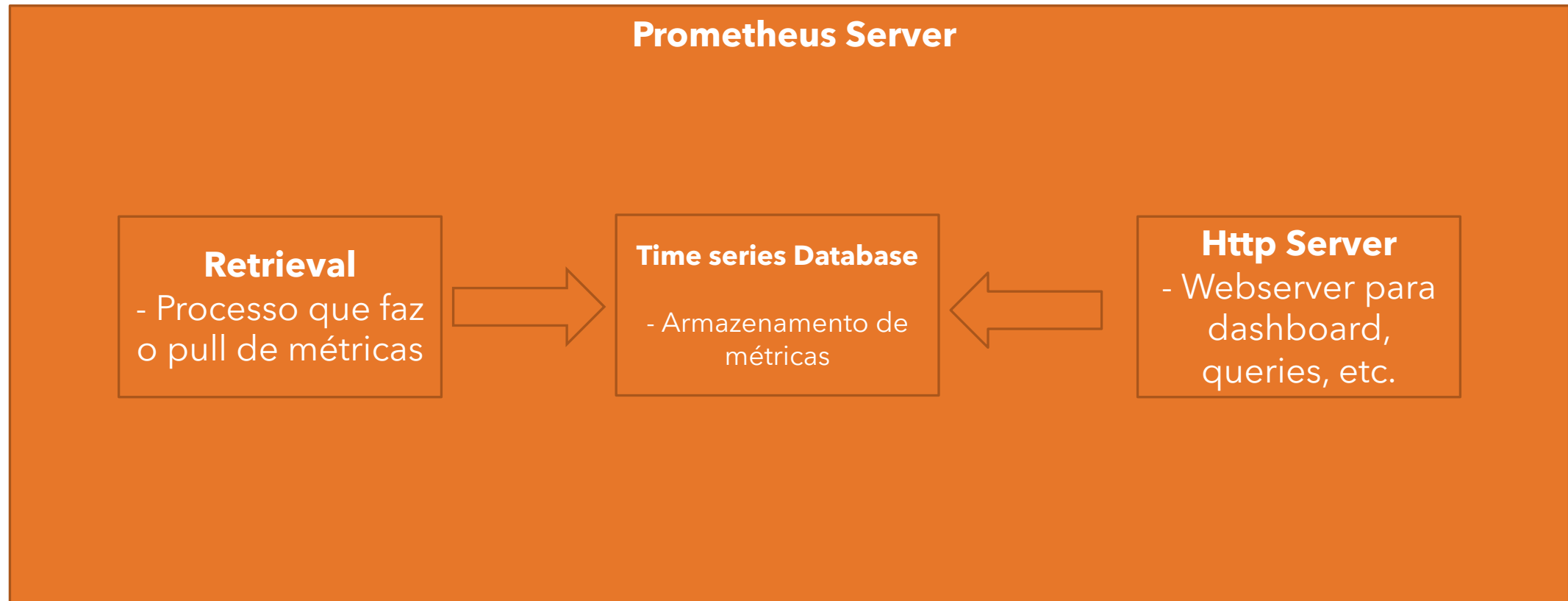
- Complexidade de DevOps cada vez mais complexa
- Mais processos, distribuídos por servidores (físicos e virtuais) containers etc.
- Como monitorar latência, falha de hardware, erros?
- Tempo de troubleshooting pode ser lento de acordo com a complexidade do ambiente.
- Monitorar constantemente toda infraestrutura
- Alertar quando algum problema ocorre imediatamente



Prometheus

# Arquitetura

---



# Arquitetura

---

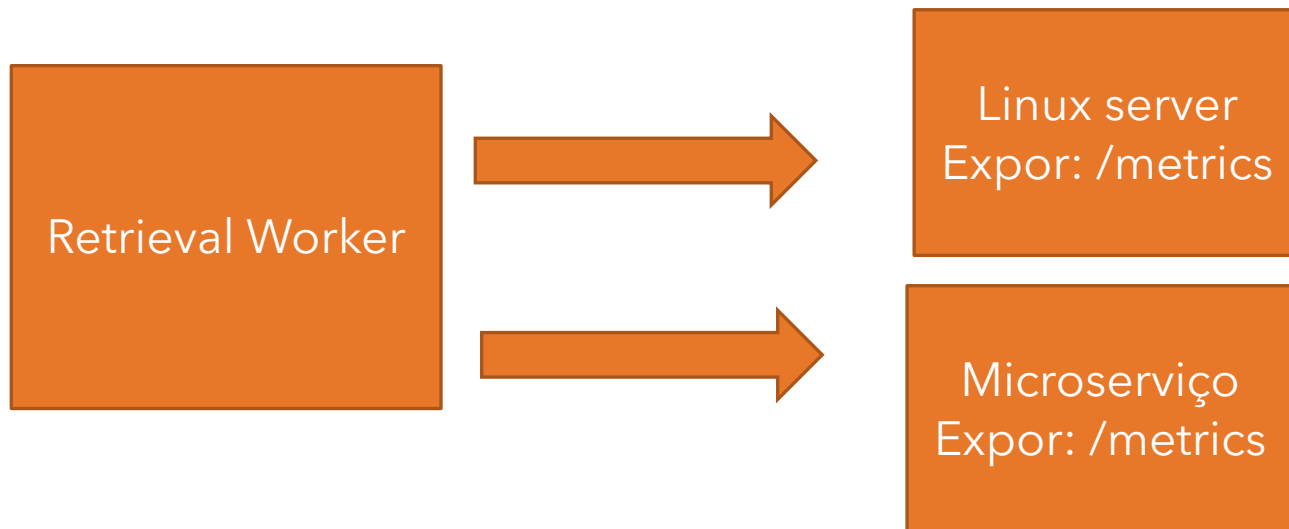
- Targets: Linux server, http server, banco de dados, microserviço, etc.
- Units: Status cpu, memória, latência, espaço em disco, exceptions, tempo de request, etc.
- Metrics: units armazenadas no Prometheus



# Arquitetura

---

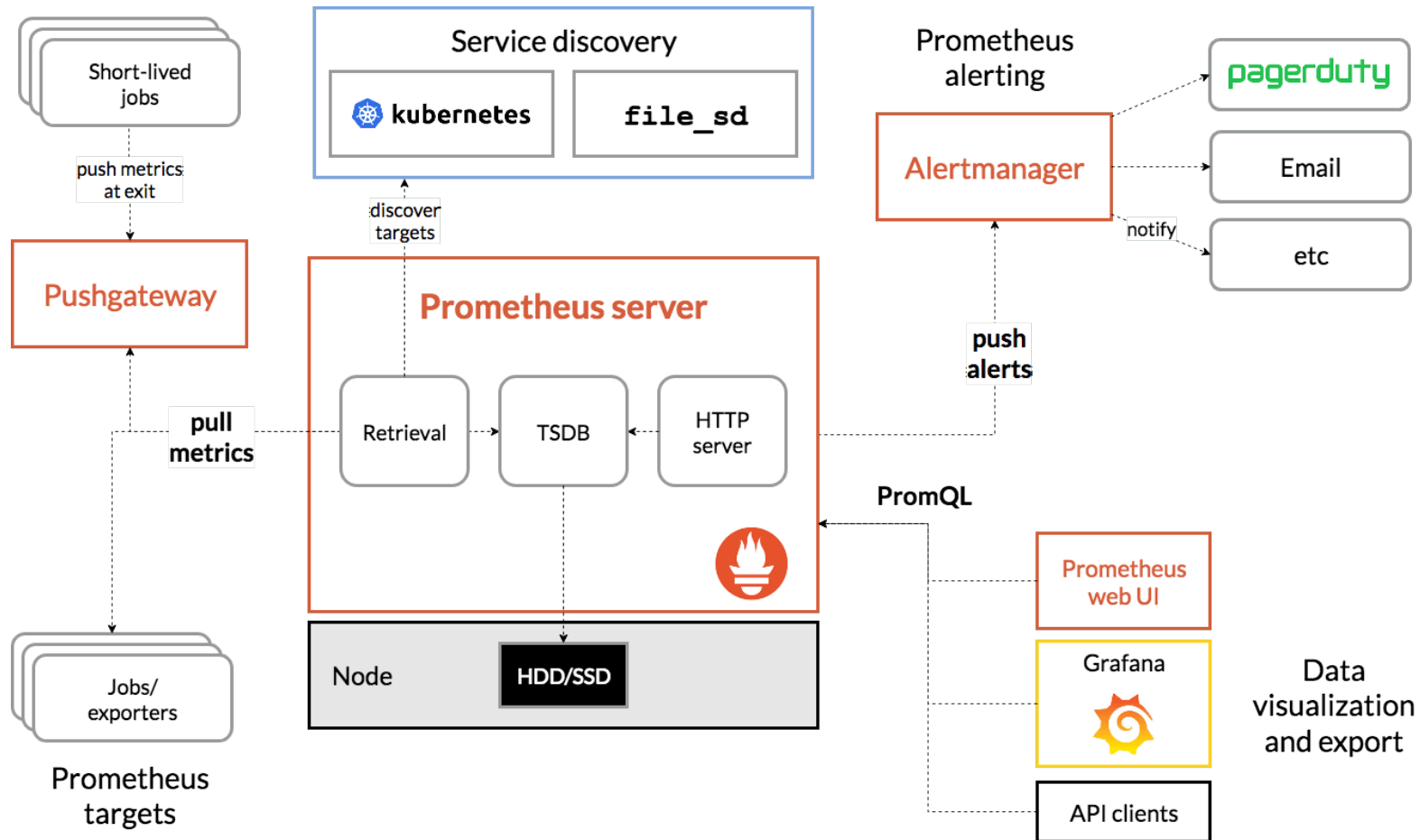
- Funcionamento do retrieval worker



- Alguns serviços já possuem /metrics expostos
- Alguns casos é preciso implementar (exporter)



# Arquitetura



# Prometheus – Mão na massa

---

- Criar VM via vagrant
  - Expor porta 9090
  - Instalação do docker via provision.sh
  - Criar arquivo de configuração prometheus.yml
- Instalação do Prometheus - docker
  - `docker run -d -p 9090:9090 -v /vagrant/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus`
- Monitorar Servidor Linux
  - [https://prometheus.io/download/#node\\_exporter](https://prometheus.io/download/#node_exporter)
  - [node\\_exporter-1.0.0.linux-amd64.tar.gz](https://github.com/prometheus/node_exporter/releases/download/v1.0.0/node_exporter-1.0.0.linux-amd64.tar.gz)
  - Extrair para /opt executar: `nohup ./node_exporter &`





# Dashboards com Grafana

---

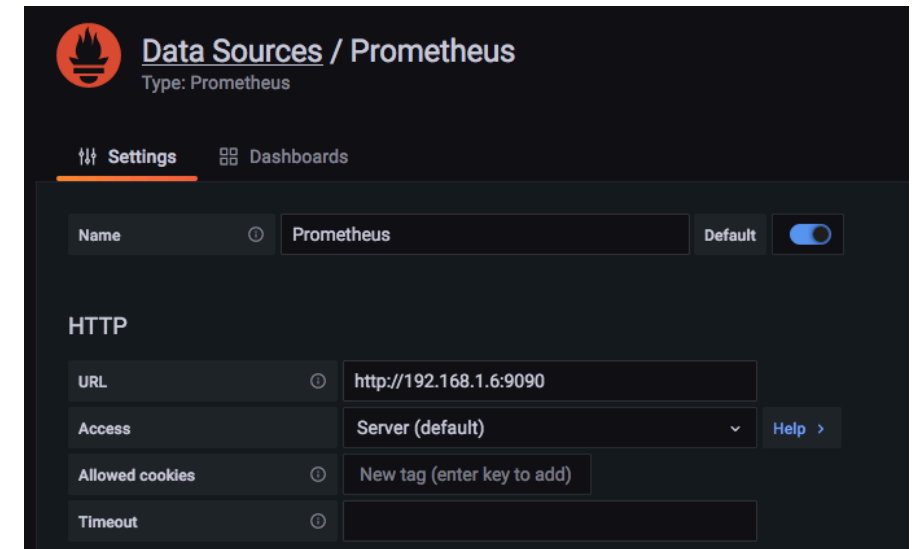
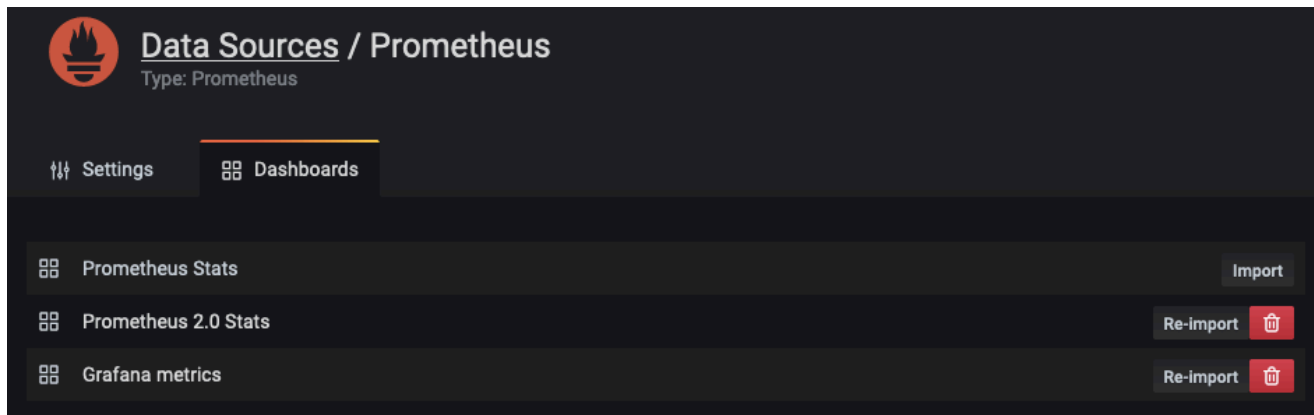
- Ferramenta open source para visualização de dados
- Suporte a criação de dashboards, queries, alertas, etc.
- Integração com Prometheus
- Dashboards pré-configurados para Prometheus



# Instalação / Configuração do Grafana

---

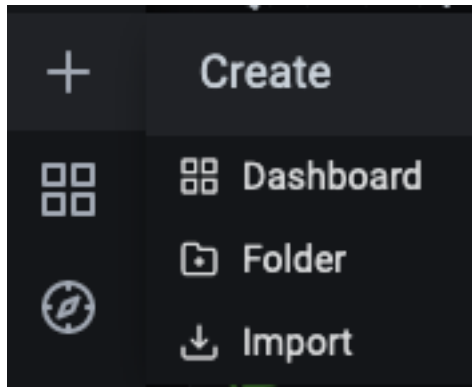
- `docker run -d -p 3000:3000 --name grafana grafana/grafana:latest`
- Adicionar datasource Prometheus
- Adicionar dashboards: 2.0 Stats e Grafana metrics



# Dashboards – node exporter

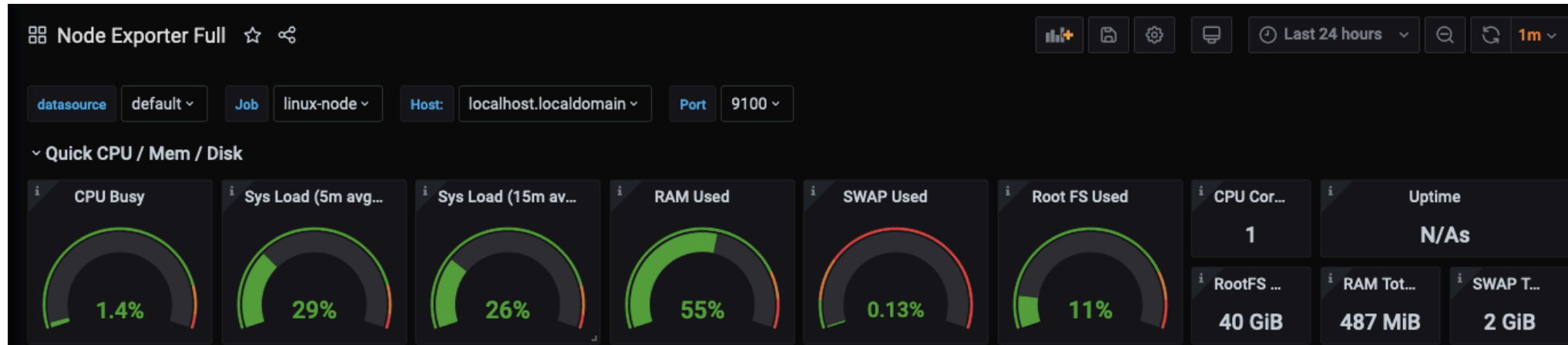
---

- Acessar repo: <https://github.com/rfrail3/grafana-dashboards/tree/master/prometheus>
- Importar dashboard node-exporter: <https://raw.githubusercontent.com/rfrail3/grafana-dashboards/master/prometheus/node-exporter-full.json>
- Botão + -> Import



# Dashboards – node exporter

---



# Teste de stress

---

- Validar consumo dos dados via agent
- Instalar pacote epel: `yum install epel-release`
- Instalar pacote stress: `yum install stress`
- Executar teste: `stress --cpu 8 --io 4 --vm 2 --vm-bytes 128M --timeout 30s`
- Validar o dashboard