Roteiro de Atividade Prática

Nome: Turma: .

**Atividade: Configurando um sistema avançado de monitoramento**

**Tempo estimado:** 35 minutos.

**Lista de materiais**

* Computador com internet;
* Caderno para anotações;
* 1 caneta,

**Objetivo:** configurar um sistema de monitoramento completo utilizando Prometheus e Grafana, integrando também alertas avançados e dashboards personalizados para visualizar e analisar métricas do pipeline.

**Enunciado**

Você está encarregado de configurar um sistema de monitoramento para um pipeline de CI/CD usando Prometheus e Grafana. A empresa exige que o sistema forneça métricas detalhadas e alertas para qualquer anomalia no pipeline, incluindo tempos de execução, taxas de falhas, uso de recursos e notificações por e-mail e Slack.

Situação fictícia elaborada especialmente para o curso.

**Passo a passo**

1. **Preparação do ambiente:**

* **Instalação do Prometheus**

Baixe e instale o Prometheus conforme a documentação oficial.

Configure o Prometheus para coletar métricas do Jenkins ou da ferramenta de CI/CD que você está usando, adicionando as métricas necessárias no prometheus.yml.

scrape\_configs:

- job\_name: 'jenkins'

scrape\_interval: 15s

static\_configs:

- targets: ['localhost:8080']

1. **Instalação e configuração do Grafana:**

* **Instalar Grafana**

Siga as instruções do site oficial.

* **Configurar conexão com Prometheus**

No Grafana, vá para Configuration > Data Sources e adicione o Prometheus com a URL http://localhost:9090.

1. **Criação de dashboards personalizados:**

Crie um novo dashboard no Grafana e adicione gráficos com as seguintes métricas:

* **Tempo de execução por etapa**

Query: avg(jenkins\_stage\_duration\_seconds)

* **Taxa de sucesso e falha**

Query: rate(jenkins\_build\_result\_total{result="SUCCESS"}[5m])

Query: rate(jenkins\_build\_result\_total{result="FAILURE"}[5m])

* **Uso de recursos do servidor**

Métricas: node\_cpu\_seconds\_total, node\_memory\_Active\_bytes, node\_disk\_io\_time\_seconds\_total

1. **Configuração de alertas:**

Configure alertas no Prometheus para notificar a equipe sobre problemas críticos:

* Exemplo de alerta no prometheus.yml:

alerting:

alert\_remanager\_config:

alertmanager\_configs:

- static\_configs:

- targets:

- 'alertmanager:9093'

rule\_files:

- 'alerts.yml'

* Crie um arquivo alerts.yml com regras de alerta, por exemplo:

groups:

- name: pipeline\_alerts

rules:

- alert: PipelineFailure

expr: rate(jenkins\_build\_result\_total{result="FAILURE"}[5m]) > 0

for: 5m

labels:

severity: critical

annotations:

summary: "Pipeline failure detected"

description: "Pipeline build has failed more than 5 times in the last 5 minutes."

1. **Integração com notificações:**

Configure o **Alertmanager** para enviar notificações por e-mail e Slack, ajustando o arquivo alertmanager.yml:

global:

smtp\_smarthost: 'smtp.gmail.com:587'

smtp\_from: 'your-email@gmail.com'

smtp\_auth\_username: 'your-email@gmail.com'

smtp\_auth\_password: 'your-password'

route:

group\_by: ['alertname']

receiver: 'slack-notifications'

receivers:

- name: 'slack-notifications'

slack\_configs:

- api\_url: 'https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXX'

channel: '#alerts'

**Procedimento experimental**

1. Após configurar o sistema, identifique e descreva em um relatório os resultados obtidos no Grafana em relação às métricas do pipeline.
2. Envio o relatório no AVA.

|  |
| --- |
|  |
|  |
|  |
|  |