ניטוק אחרי מודלים בשרת טרייטון עם Grafana & Prometheus:

הורדה והרצה

קונפיגורציה עם שרת פרומסיוס

הכנת שרת GRAFANA:

1ץ.

בשביל להריץ גרפנה נשתמש בדוקר:

docker run -p 3000:3000 (-v Grafana-storage:/var/lib/Grafana) grafana/grafana

2. התקנת Grafana (wget & dpkg -i name

3. הרצת גרפנה –

./bin/grafana-server

מאזין בפורט 3000

הסיסמה והמשתמש זה admin

קישור לפרומיסיוס סרבר:

בחירה ב add data source

בחירה בפרומיסיוס

נתינת שם והכנסת url של ROUTE סלאש METRICS

<https://github.com/ChristianLempa/boilerplates/tree/main/docker-compose/prometheus>

**אפשרות 1: התקנה נפרדת של Prometheus באמצעות Prometheus Operator**

**1. התקנת Prometheus Operator**

* היכנסו ל-OpenShift באמצעות כלי השורה oc.
* נווטו ל-OperatorHub וחפשו את "Prometheus Operator".
* לחצו על "Install" ופעלו לפי ההוראות להתקנה.

**2. יצירת ServiceMonitor עבור Triton**

ServiceMonitor הוא רכיב של Prometheus Operator שמגדיר כיצד Prometheus צריך לגלות ולנטר את ה-Service של Triton.

צרו קובץ YAML בשם triton-service-monitor.yaml עם התוכן הבא:

YAML

apiVersion: monitoring.coreos.com/v1

kind: ServiceMonitor

metadata:

name: triton-monitor

namespace: <your-prometheus-namespace> # החליפו ב-Namespace בו פועל Prometheus

spec:

selector:

matchLabels:

app: triton # התאימו לתווית של ה-Service של Triton

endpoints:

- port: metrics # שם הפורט של המדדים ב-Service של Triton

path: /metrics # הנתיב endpoint של ה-metrics

interval: 15s # באיזו תדירות לאסוף מדדים

* **namespace:** ודאו שזה תואם ל-Namespace בו פועל Prometheus Operator.
* **selector:** התאימו את התוויות (matchLabels) כך שיתאימו לתוויות של ה-Service של Triton שמכיל את פורט 8002.
* **port:** זה צריך להתאים לשם הפורט של המדדים ב-Service של Triton (במקרה שלכם, זה כנראה metrics).
* **path:** הנתיב endpoint של ה-metrics. בדרך כלל /metrics.
* **interval:** מגדיר את תדירות איסוף המדדים.

החילו את הקובץ:

Bash

oc apply -f triton-service-monitor.yaml

**3. בדיקה**

* בדקו את הלוגים של Prometheus Operator כדי לוודא שהוא גילה את ה-Service של Triton.
* בדקו את ה-Targets של Prometheus (דרך ה-UI של Prometheus) כדי לוודא ששרת ה-Triton מופיע שם.
* חפשו מדדים של Triton ב-Prometheus UI.

**אפשרות 2: הוספת Prometheus לפריסה הנוכחית של Triton (פחות מומלץ)**

אפשרות זו מתאימה אם אתם רוצים להפעיל את Prometheus יחד עם שרת ה-Triton באותו Pod.

**1. הוספת Prometheus כקונטיינר נוסף**

ערכו את קובץ ה-Deployment שלכם והוסיפו קונטיינר נוסף עבור Prometheus.

YAML

spec:

template:

spec:

containers:

- name: triton

image: <your-triton-image>

ports:

- containerPort: 8000

- containerPort: 8001

- containerPort: 8002

- name: prometheus

image: prom/prometheus:latest

ports:

- containerPort: 9090

args:

- '--config.file=/etc/prometheus/prometheus.yml'

volumeMounts:

- name: prometheus-config

mountPath: /etc/prometheus/prometheus.yml

subPath: prometheus.yml

volumes:

- name: prometheus-config

configMap:

name: prometheus-config

**2. יצירת ConfigMap עבור Prometheus**

צרו ConfigMap עם קובץ קונפיגורציה עבור Prometheus.

YAML

apiVersion: v1

kind: ConfigMap

metadata:

name: prometheus-config

data:

prometheus.yml: |

global:

scrape\_interval: 15s

scrape\_configs:

- job\_name: 'triton'

static\_configs:

- targets: ['localhost:8002']

**3. הפעלה מחדש של הפריסה**

הפעילו מחדש את הפריסה כדי שהשינויים יחולו.

**הערות**

* אפשרות 1 (התקנה נפרדת) מומלצת יותר, מכיוון שהיא מאפשרת ניהול נפרד של Prometheus ושרת ה-Triton.
* בשתי האפשרויות, ודאו שה-Route etty-metrics.com אכן מפנה לפורט 8002 של ה-Service של Triton.
* ייתכן שתצטרכו להתאים את ההגדרות בהתאם לסביבה שלכם.

**סיכום**

שתי הגישות שהוצגו מאפשרות לכם להוסיף את Prometheus לשרת ה-Triton שלכם ב-OpenShift. התקנה נפרדת באמצעות Prometheus Operator היא הגישה המומלצת, מכיוון שהיא פשוטה יותר ומאפשרת ניהול נפרד של Prometheus.

בואו נפרט כיצד לבדוק את ה-Targets של Prometheus וכיצד לראות את התוצאות של Prometheus בשתי האפשרויות שהזכרנו.

אפשרות 1: התקנה נפרדת של Prometheus באמצעות Prometheus Operator

בדיקה של Targets

גישה לממשק המשתמש של Prometheus: פתחו את הדפדפן שלכם והזינו את כתובת ה-URL של Prometheus. כתובת זו תלויה באופן שבו התקנתם את Prometheus, אך לרוב זה יהיה משהו כמו http://<prometheus-server-address>:9090.

לשונית "Status": בתפריט העליון של Prometheus, לחצו על הלשונית "Status".

תפריט "Targets": בתפריט המשני, לחצו על "Targets".

רשימת Targets: תראו רשימה של כל ה-Targets ש-Prometheus עוקב אחריהם. חפשו את ה-Target שמתאים לשרת ה-Triton שלכם. זה עשוי לכלול את שם ה-Service של Triton ואת ה-Namespace שלו.

מצב Target: ודאו שהסטטוס של ה-Target הוא "Up". זה מציין ש-Prometheus מצליח לתקשר עם שרת ה-Triton ולקבל ממנו מדדים.

בדיקה של מדדים

לשונית "Graph": בתפריט העליון של Prometheus, לחצו על הלשונית "Graph".

שאילתות PromQL: בתיבת הטקסט, הקלידו שאילתה בשפת PromQL כדי לראות את המדדים של Triton. לדוגמה, כדי לראות את מספר הבקשות שהתקבלו, תוכלו להשתמש בשאילתה nv\_inference\_request\_count.

ביצוע שאילתה: לחצו על "Execute" כדי להריץ את השאילתה.

תצוגה: Prometheus תציג את התוצאות בצורה גרפית או טבלאית.