

گزارش عملکرد اولیه

موضوع: مقایسه عملکرد Docker و Podman در اجرای کانتینرها

درس: پروژه کارشناسی

نام دانشجو: محمد مهدی رسول امینی

نام استاد: استاد سلطانی

تاریخ: 1404/02/26

مقدمه

در دنیای زیرساخت ابری و مجازی سازی، ابزارهای مدیریت کانتینر نقش کلیدی در توسعه و استقرار سرویس ها دارند. Docker، به عنوان پرکاربردترین ابزار کانتینر، استاندارد برای بسته بندی و اجرای نرم افزارها ارائه داده است. Podman نیز به عنوان جایگزینی سبک تر و امن تر برای Docker معرفی شده است. در این گزارش عملکرد این دو ابزار در اجرای کانتینرها مقایسه می شود.

تست عملکرد Docker

محیط آزمایش:

- سیستم عامل: Kali linux 2024
- نسخه Docker: 24.0.5
- ابزار تست: sysbench

تست CPU با: sysbench

docker run --rm alpine sysbench cpu --cpu-max-prime=20000 run

نتایج:

- سرعت پردازش: 216.65 event/sec
- زمان کل: 10.002 ثانیه

- Latency:

- کمینه: ms1.99
- میانگین: ms4.61
- بیشینه: ms22.91
- صدک ۹۵: ms7.19

تست حافظه با: sysbench

```
docker run --rm alpine sysbench memory --memory-block-size=1K --memory-total-size=10G run
```

نتایج:

- سرعت انتقال داده: MiB/sec2864.69
- عملیات کل: 10485760 بار
- Latency:
 - میانگین: ms0.00
 - بیشینه: ms4.95

۳. ویژگی‌های Docker

- ساختار Client-Server
 - سادگی در کاربری و یکپارچه با ابزارهای CI/CD
 - پشتیبانی وسیع از اکوسیستم ابری
-

تست عملکرد Podman

مقدمه:

Podman به عنوان جایگزین daemon-less برای Docker طراحی شده که قابلیت‌های مشابه را بدون نیاز به سرویس پس‌زمینه فراهم می‌کند.

تست CPU با: sysbench

```
podman run --rm docker.io/library/alpine sysbench cpu --cpu-max-prime=20000 run
```

نتایج:

- سرعت پردازش: 215.80 event/sec
- زمان کل: 10.002 ثانیه
- Latency:
 - کمینه: 2.01 ms
 - میانگین: 4.63 ms
 - بیشینه: 23.10 ms
 - صدک ۹۵: 7.20 ms

تست حافظه با: sysbench

```
podman run --rm docker.io/library/alpine sysbench memory --memory-block-size=1K --memory-total-size=10G run
```

نتایج:

- سرعت انتقال داده: 2840.12 MiB/sec
- عملیات کل: 10485760 بار
- Latency:
 - میانگین: 0.00 ms
 - بیشینه: 5.10 ms

ویژگی‌های Podman:

- ساختار **daemon-less** و اجرا به صورت **rootless**
- امنیت بیشتر در محیط‌های چندکاربره
- سازگاری کامل با Dockerfile و تصاویر Docker
- طراحی شده بر اساس کتابخانه **libpod**

مقایسه اولیه Podman و Docker

ابزار	سرعت CPU (event/sec)	سرعت حافظه (MiB/sec)	ساختار	نیاز به Daemon	امنیت Rootless	سازگاری با Dockerfile
Docker	216.65	2864.69	Client- Server (Daemon- based)	دارد	محدود	بله
Podman	215.80	2840.12	Daemon- less	ندارد	بله	بله

تحلیل:

- عملکرد CPU و حافظه در هر دو ابزار بسیار نزدیک به هم است.
 - تفاوت‌های جزئی ناشی از معماری اجرا و سربار سیستم است.
 - در محیط‌هایی که امنیت و عدم وابستگی به daemon اهمیت دارد، Podman مزیت بیشتری دارد.
 - برای پروژه‌های سریع و سازگار با اکوسیستم DevOps، Docker همچنان محبوب‌تر است.
-

نتیجه‌گیری تا این مرحله:

1. در تست‌های اولیه تفاوت چشمگیری از نظر عملکردی بین Docker و Podman مشاهده نشد.
2. انتخاب بین این دو ابزار بیشتر به نیازمندی‌های معماری، امنیتی و ساختاری وابسته است.
3. ابزارهای تستی مانند stress-ng و sysbench به راحتی در هر دو محیط اجرا شدند و نتایج مشابهی ارائه دادند.