به نام خدا

گزارش فاز اول پروژه

نویسنده: محمدحسن بیاتیانی
شماره دانشجویی: ۴۰۱۱۰۵۶۹۱

۱. شرح مفاهیم

در این بخش به معرفی مفاهیم کلیدی مورد استفاده در پروژه می‌پردازیم که پایه و اساس ساخت یک سیستم مدیریت محفظه در لینوکس هستند.

۱.۱ Container (محفظه)

کانتینر یک فناوری مجازی‌سازی سبک‌وزن است که به کمک آن می‌توان محیطی ایزوله برای اجرای برنامه‌ها ایجاد کرد، بدون اینکه نیاز به اجرای سیستم‌عامل کامل جدید باشد. برخلاف ماشین‌های مجازی که هرکدام دارای هسته جداگانه هستند، کانتینرها از همان هسته‌ی سیستم‌عامل میزبان استفاده می‌کنند ولی با استفاده از قابلیت‌هایی مانند namespace و cgroup، از یکدیگر و از میزبان جدا می‌شوند.

مزایای اصلی کانتینرها:

- سبک‌تر و سریع‌تر از ماشین‌های مجازی
- راه‌اندازی سریع
- قابل حمل بودن (portability)
- تکرارپذیری بالا در اجرای برنامه‌ها
- مناسب برای توسعه، تست و استقرار مداوم (CI/CD)

۱.۲ Namespace

Namespaceها یکی از قابلیت‌های اصلی هسته لینوکس هستند که امکان ایزوله‌سازی بخش‌های مختلف سیستم‌عامل برای هر فرآیند را فراهم می‌کنند. انواع مختلفی از namespaceها در لینوکس وجود دارند:

- **UTS namespace**: امکان تعیین hostname و domain name جداگانه برای هر محفظه
- **PID namespace**: محفظه دارای جدول PID اختصاصی است

- **Mount namespace**: دید جداگانه از فایل‌سیستم
- **Network namespace**: استک شبکه مجزا برای هر محفظه
- **IPC namespace**: ایزوله‌سازی در مکانیزم‌های IPC
- **User namespace**: اجازه دسترسی root ایزوله‌شده برای کاربر عادی

۱.۳ Control Groups

Cgroupها برای کنترل و محدودسازی مصرف منابع توسط گروهی از فرآیندها به‌کار می‌روند.

- **Memory**: تعیین محدودیت مصرف رم
- **CPU**: تخصیص زمان پردازنده
- **Block I/O**: کنترل سرعت دیسک
- **Freezer**: توقف یا ادامه اجرای گروهی از پردازه‌ها

۱.۴ Chroot

Chroot روشی برای تغییر مسیر ریشه فایل‌سیستم یک فرآیند است. این کار منجر به ایزوله‌سازی نسبی فایل‌ها برای پردازه‌های داخل آن مسیر می‌شود. اگرچه امنیت بالایی ندارد، اما به‌عنوان ابزاری مکمل در ایزوله‌سازی فایل‌سیستمی استفاده می‌شود.

۱.۵ Union Filesystem

Union filesystem (مانند OverlayFS) لایه‌های مختلف فایل‌سیستم را به‌صورت پویا ترکیب می‌کند. یک لایه فقط‌خواندنی (مانند سیستم پایه) با لایه فقط‌نوشتنی ترکیب شده و محیطی برای اجرای برنامه‌ها فراهم می‌شود.

۱.۶ eBPF

eBPF فناوری جدید و پیشرفته‌ای در کرنل لینوکس است که امکان اجرای برنامه‌هایی را در سطح هسته بدون تغییر کد هسته فراهم می‌کند. در این پروژه می‌توان از eBPF برای نظارت بر system callها، رفتار منابع و تحلیل عملکرد محفظه‌ها استفاده کرد.

ویژگی‌های eBPF:

- اجرای امن و سریع در سطح هسته
- بدون نیاز به ماژول کرنل
- مناسب برای مانیتورینگ دقیق، امنیت و فیلتر کردن داده‌ها در شبکه و سیستم

۲. معرفی ابزارها و محیط اجرایی

۲.۱ ابزارهای مورد استفاده

- **ابزارهای لینوکس:**
  unshare, ip, mount, cgcreate, chroot, ps, top
- **زبان برنامه‌نویسی:**
  Bash (در فاز دوم ممکن است Python یا C استفاده شود)
- **ابزارهای مرجع:**
  Podman – به‌عنوان یک runtime سبک و بدون daemon
- **سیستم فایل:**
  OverlayFS برای پیاده‌سازی فایل‌سیستم چندلایه
- **ابزارهای مدیریت منابع:**
  cgroup-tools برای ساخت و استفاده از گروه‌های کنترل

۲.۲ محیط اجرایی

- **سیستم عامل:** Ubuntu
- **نسخه کرنل:** 5.4 یا بالاتر با پشتیبانی از namespace و cgroup
- **دسترسی root:** برای اعمال دستورات سطح پایین مثل mount و chroot
- **پیش‌نیازها:** نصب پکیج‌هایی مثل cgroup-tools, util-linux, overlayfs-tools