

اگر در طول سال گذشته هر مقدار از وقت خود را صرف مطالعه "open source" کرده باشید، احتمالاً درباره docker شنیده اید. این روزها بسیاری از مردم در مورد docker و تاثیر آن در مجازی سازی و DevOps صحبت می کنند .



وقتی تکنولوژی جدیدی مانند docker ظهور میکند معمولاً میتواند سخت باشد که تبلیغات افراطی و هیجان برانگیز را نادیده بگیریم و پیامدهای عملی موضوع را درک کنیم. علاوه بر این، وجود اصطلاحات پیچیده در موضوعاتی مانند "کانتینرهای لینوکس" اغلب افراد غیرمتخصص را گیج میکند، و بحث و گفتگوی مربوطه را فقط به کسانی که اطلاعات عمیقی در این زمینه دارند، محدود میکند. با استفاده از این مقاله، من لحظه ای از بحث درباره آنچه docker میتواند انجام دهد، عقب میکشم تا بتوانم بر این که docker چگونه در حال تغییر چشم انداز لینوکس است، تمرکز کنم.

داگر دقیقاً چیست؟

به طور خلاصه، docker توسعه ی روش "کانتینرهای لینوکس" است: یک نوع منحصر به فرد از یک مجازی سازی بسیار سبک و نرم افزار محور است که به شدت سربارها را کاهش میدهد و برنامه نویسی در سرورها را آسان تر میکند. "سالومون هیکس"، بنیانگذار داکر، برای توضیح این قابلیت، آن را با کانتینرهای حمل و نقل استاندارد که برای حمل کالاهای متنوع به سراسر جهان استفاده می شوند، مقایسه می کند Docker. به مدیران سیستم ها و سازندگان این اجازه را می دهد تا نرم افزارهایی بسازند که بتواند بر روی همه محصولات لینوکس یا هر سخت افزار مجازی شده

بدون آن که نیاز به ایجاد ساختارهای سفارشی برای محیطهای مختلف داشته باشد، اجرا شود. این ویژگیها برندهای بزرگی را به خود جذب میکند و docker را یکی از موفقترین پروژههای open source سال گذشته تبدیل کرده است. با این شرایط به نظر می رسد که docker قرار است ماندگار شود. حال این چه معنایی برای لینوکس دارد؟



کاربردهای بسیار Docker

شرکت Red Hat در راس توسعه دادن و به کارگیری docker قرار داشته، و "پاول کورمیر" یکی از بزرگترین طرفداران استفاده از آن بوده است. این شرکت از ماه سپتامبر سال گذشته با docker همکاری نزدیکی داشته، و بهبود عملکرد docker در پلتفرم openshift تمرکز کرده است. تمرکز کلی روی استفاده از Docker به عنوان یک ابزار مکانیکی برای بهتر کردن مدیریت منابع، انعطاف فرآیند، و امنیت در مجازی سازی برنامه بوده است. این تلاشها با راه اندازی پروژه Atomic (یک میزبان لینوکس سبک که به طور خاص برای اجرای کانتینرهای لینوکس طراحی شده است) به سرانجام رسیده اند. هدف این پروژه این است که کانتینرها، به آسانی گسترش یابند، به روز رسانی شوند، و به محیطی که نیاز به منابع بسیار کمتری نسبت به یک میزبان معمولی لینوکس دارد، بازگردانده شوند.

داکر برای DevOps

یکی دیگر از نکات مهم در کاربرد Docker نقش آن در جامعه DevOps است. Docker به گونه ای طراحی شده است که می توان آن را در اکثر برنامه های DevOps از جمله Vagrant, Chef, Puppet و Ansible گنجاند، و یا از آن به تنهایی برای مدیریت محیط های سازندگی استفاده کرد. علت اصلی فروش این است که بسیاری از وظایفی که

معمولا توسط برنامه های دیگر انجام می شود را ساده می کند. به بیان دقیق تر، Docker این امکان را فراهم می سازد که محیط های توسعه محلی ای را ایجاد کنیم که دقیقا همانند یک سرور زنده است، یا امکان اجرای محیط های توسعه چندگانه از یک میزبان را فراهم می کند که هر یک از این محیط ها دارای نرم افزار، سیستم عامل و پیکربندی منحصر به فردی هستند، همچنین میتواند امکان آزمایش پروژه ها در سرور های جدید یا متفاوت را فراهم کند و اجازه می دهد هر کسی در همان پروژه با همان تنظیمات، بدون در نظر گرفتن محیط میزبان محلی فعالیت کند. در آخر، برای یک تیم سازنده، Docker می تواند نیاز به این که نسخه های مشابهی از همه چیز را بر روی دستگاه محلی خود نصب کنند، از بین ببرد.

Spotify در حال کار بر روی ترکیب Docker در توسعه جریان کار خود است. ماهیت تکرارپذیری تصاویر Docker، استاندارد سازی کد تولید و تنظیمات را برای آن ها آسان تر می کند.

فعالیت آن ها منجر به ایجاد Helios شده است، یک برنامه کاربردی که گسترش Docker را در سرورهای مختلف مدیریت می کند و هنگامی که یک سرور نسخه صحیح یک کانتینر را اجرا نمیکند، هشدار می دهد.

داکر برای ادغام پیوسته

eBay متمرکز شده است تا Docker را به فرآیند ادغام پیوسته خود متصل کند تا نصب در یک شبکه توزیع شده از سرورها که به عنوان یک خوشه واحد اجرا می شود، را استاندارد سازی کند. آن ها وابستگی های کاربردی را در کانتینر ها جداسازی می کنند تا به مسئله نسخه های مختلف نرم افزار هر سرور، پاسخ دهند. به این معنا که سیستم عامل میزبان نیازی ندارد دقیقا همانند سیستم عامل کانتینری باشد، و هدف نهایی آن ها این است که سیستم های مختلف سخت افزاری و نرم افزاری را به عنوان یک خوشه Mesos اجرا کنند.

Docker برای امنیت سند باکس ها

مصاحبه از راه دور (remote interview) نرم افزاری کاربردی برای استخدام کنندگان فراهم آورده که با آن مهارت های نامزد های شغلی را از راه دور می سنجند. آن ها CompileBox را منتشر کردند، یک سندباکس مبتنی بر Docker که می تواند کدهای غیر قابل اعتماد را اجرا کند و بدون به خطر انداختن میزبانی که در حال اجرای نرم افزار است، خروجی را بازگرداند. در مراحل توسعه CompileBox، تیم حاضر در پروژه مصاحبه از راه دور در نظر داشتند که از Chroot jails, Ideone, یا ماشین های مجازی سنتی استفاده کنند، ولی سرانجام به این نتیجه رسیدند که docker بهترین گزینه است Chroot. سطح امنیتی مورد نیاز را فراهم نمی کند، Ideone میتواند سریع هزینه برانگیز شود، و ماشین های مجازی پس از از بین رفتن اطلاعات زمان فوق العاده طولانی برای راه اندازی مجدد نیاز دارند.

Docker انتخابی آشکار برای این برنامه بود؛ زیرا کد مخرب که تلاش می کند سیستم را از بین ببرد، محدود به کانتینر است و کانتینرها در صورت نیاز می توانند به سرعت ایجاد و تخریب شوند.