فصلنامه رویکردهای نوین در تحقیقات علوم پایه، فنی و مهندسی
سال پنجم، شماره ۱۷، تابستان ۱۴۰۱
ISSN: ۲۶۴۵-۵۴۳۹
www.jonarbset.ir

تجزیه و تحلیل داکر و معماری آن و مقایسه آن با ماشین مجازی

بتول عادلی [۱] ، محمدکاظم بشکنی [۲]

[۱] کارشناسی مهندسی کامپیوتر-دانشگاه آزاد اسلامی واحد تبریز
[۲] کارشناسی کامپیوتر-موسسه آموزش عالی حکیم نظامی (نویسنده مسئول)

چکیده

بصورت کلی Docker یک محصول Open Source است که فرایند ایجاد ، توسعه و اجرا کردن Application ها با استفاده از Container ها در قسمت OS Level Virtualization را بسیار ساده می کند .Container ها به یک برنامه نویس این اجازه را می دهد که application خود را با تمامی اجزای مورد نیاز آن اعم از فایل های dll و کلیه library ها ، کلیه وابستگی ها در قالب یک بسته نرم افزاری ارائه بدهد به شکلی که از بیرون یک نرم افزار واحد به نظر برسد. داکر پلتفرمی است برای ساخت، اجرا انواع اپلیکیشن ها مدیریت کانتینرها مورد استفاده قرار می گیرند. یکی از مهم ترین مفاهیم در داکر کانتینر است که با استفاده از آن به مراتب ساده تر از ماشین های مجازی می تواند نرم افزار های خود را روی پلتفرم های مختلفی اجرا کند . داکر علاوه بر لینوکس روی سیستم عامل ویندوز و سایر سیستم عامل های دیگر قابل اجرا است. داکر یکی از موفق ترین پروژه های متن باز در تاریخ فناوری اطلاعات است. که توسعه های نرم افزارهای کاربردی را درون کانتینر نرم افزاری به وسیله فراهم کردن لایه انتزاعی اضافه ای فراهم می کند.سازمان ها همواره در تلاش برای افزودن قابلیت حمل برنامه های کاربردی خود از طریق کانتینرها هستند.فرآیند استقرار نرم افزارها و سرویس ها رو با معرفی مفهوم کانتینرها سرعت می بخشد .در این مقاله به طورکلی مجازی سازی با داکر کانتینر مورد بررسی قرار گرفته است.

واژه‌های کلیدی: داکر، ماشین مجازی، کانتینر،SWARM

۱- مقدمه

مجازی سازی مبتنی بر کانتینر ، جایگزینی سبک وزن برای ابر ناظرها است و به عنوان مجازی سازی سطح سیستم عامل نیز شناخته می شود. این نوع از مجازی سازی با ایجاد چند نمونه فضای مجزای کاربر، منابع ماشین فیزیکی را تقسیم می کند مجازی سازی مبتنی بر ابر ناظر در سطح انتزاع سخت افزار کار انجام می شود. از دید کاربران هر کانتینر دقیقا همانند یک سیستم عامل مستقل کار می کند. انزوا در این نوع مجازی سازی از طریق فضاهای نام namespace هسته ارائه می شود.[۱] این ویژگی با فراخوانی سیستمی قابل دسترسی است. مدیریت منابع توسط گروه های کنترل یا cgroups انجام می شود. توسط cgroups می توانیم منابعی مانند زمان پردازنده ، حافظه، سیستم پهنای باند یا ترکیبی از این منابع را در میان گروه هایی از فرآیندهایی که توسط کاربر تعریف شده اند که در سیستم اجرا می شوند تخصیص دهیم با cgroup می توان به سادگی مقیاس یک کانتینر را تغییر یا تمام فرآیندهای داخل یک کانتینر را خاتمه داد. ویژگی اصلی مجازی سازی مبتنی بر کانتینر این است که بر خلاف ماشین های مجازی سیستم عامل های کاملی را اجرا می کند.[۲]

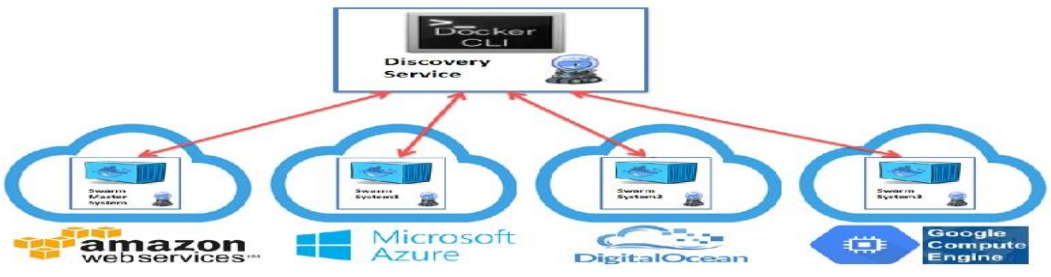
۲- ایجاد یک سیستم مجازی از سیستم ها با استفاده از DOCKER SWARM

این شبیه سازی از یک SOS مجازی docker swarm مبنی بر virtual box،Mac os X،nginx است این SOS مجازی به عنوان کلاستر چهار سیستم SWARM (vms) در virtual box روی همان کامپیوتر میزبان پیاده سازی شده است.

به هرحال ، همان SOS در هر کدام از ابر پشتیبانی روی داکر توسط تغییر پذیر نام راه انداز در شکل ۱-۲به نام ابر مطلوب چنانچه نشان داده شده در شکل ۲-۲ ایجاد می شود. [۳]

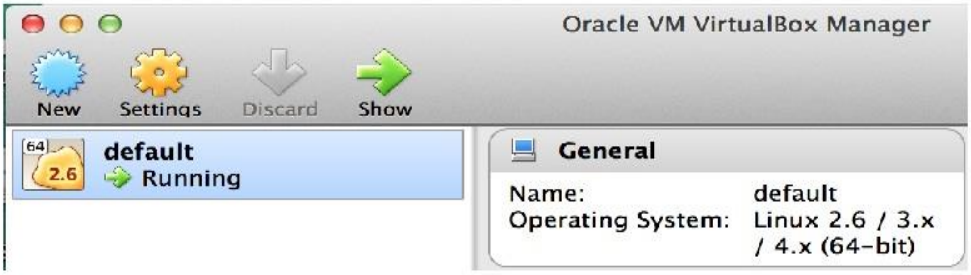
```
Nitins-MacBook-Pro:~ nitinnaik$ docker-machine create --driver=virtualbox --swarm --swarm-master
--swarm-discovery=token://1e819d13e35941894c20e4f7c8d7bcb2 my-swarm-master
Nitins-MacBook-Pro:~ nitinnaik$ docker-machine create --driver=virtualbox --swarm
--swarm-discovery=token://1e819d13e35941894c20e4f7c8d7bcb2 my-swarm-node1
Nitins-MacBook-Pro:~ nitinnaik$ docker-machine create --driver=virtualbox --swarm
--swarm-discovery=token://1e819d13e35941894c20e4f7c8d7bcb2 my-swarm-node2
Nitins-MacBook-Pro:~ nitinnaik$ docker-machine create --driver=virtualbox --swarm
--swarm-discovery=token://1e819d13e35941894c20e4f7c8d7bcb2 my-swarm-node3
```

شکل ۱-۲ با ایجاد کردن docker swarm image برای کاربردهایی به عنوان نشانه کشف سرویس



شکل ۲- ۲ سیستم مجازی docker swarm – based روی ابرهای چندگانه در معماری سیستم ها

داکر در ابتدا اجرا می کند در vm به صورت پیش فرض چنانچه در شکل ۳-۲ فراخوانی می کند ، ایجاد می کند کلاستر swarm برای اتصال ها ، اول مرحله ای است که تصویر docker swarm چنانچه نشان داده شده در شکل ۴-۲ ایجاد می کند به عنوان نشانه کشف استفاده می کنید که تمام اتصال ها در مدت کلاستر swarm متصل کند. این شبیه سازی سرویس کشف مبنی بر docker hub جزئی استفاده می شود. اکنون این شاخه image swarm استفاده می کند که تمام اتصالات swarm شامل کلاستر swarm در virtual box ایجاد کند.چنانچه نشان داده شده در شکل ۱-۲. در این sos یک swarm اصلی و اتصال سه swarm چنانچه نشان داده شده در شکل ۱-۲ ایجاد شده است. اینجا راه انداز virtual box، اوراکل استفاده شده است(همان طور که در شکل ۱-۲ می بینید). به هر حال تمام این ها اتصال های swarm در ابرهای متفاوت تغییر پذیرند روی نام راه انداز به نام ابر مطلوب از شکل ۲ در شکل ۱ موفق شده اجرا می شود.[۳]



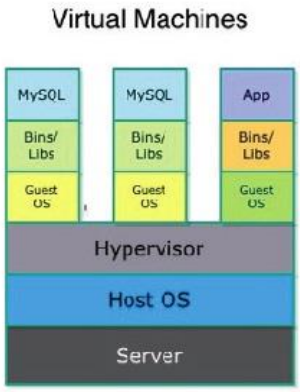
شکل ۳-۲ اجرای موتور داکر در ماشین مجازی پیش فرض

```
Nitins-MacBook-Pro:~ nitinnaik$ docker run swarm create
Unable to find image 'swarm:latest' locally
latest: Pulling from library/swarm
51436fd4bb0d: Pull complete
c31a5390266f: Pull complete
e40019be13ea: Pull complete
a3ed95caeb02: Pull complete
Digest: sha256:3add485cb6bb71c7113243753c5f484561549d9f782154b1c809219c9754ce46
Status: Downloaded newer image for swarm:latest
1e819d13e35941894c20e4f7c8d7bcb2
```

شکل ۴-۲ با ایجاد کردن تصویر docker swarm برای استفاده(کاربرد) به عنوان نشانه کشف سرویس

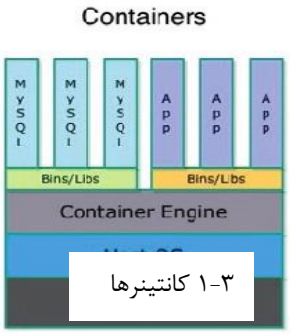
۳- ماشین های مجازی و کانتینرها

ماشین های مجازی روی یک کامپیوتر توسط hypervisor اجرا می شوند hypervisor یک مدل از مجازی سازی سخت افزاری یا hardware virtualization هست که به شما امکان اجرا از چندین سیستم عامل guest را در یک زمان روی یک سیستم host را فراهم می کند، در این حالت سیستم عامل های مجازی نصب شده همانند هر سیستم عامل واقعی امکان استفاده از منابع سخت افزاری موجود در سیستم مانند cpu و یا hard و ram را دارد. ماشین مجازی در واقع یک شبیه سازی از کامپیوتر واقعی هست که مثل یک کامپیوتر واقعی اپلیکیشن ها را اجرا می کند.[۴]



۳-۱ کانتینرها

بر خلاف ماشین های مجازی که مجازی سازی را در لایه hardware انجام می دهد سیستم های container base مجازی سازی را در لایه سیتم عامل انجام می دهند. کانتینرها و ماشین های مجازی برای تمام اهداف یکی هستند فضای اختصاصی خودشان را برای پردازش دارند و می توانند کامندها را اجرا می کنند در container base در مقابل ماشین های مجازی کانتینرها که کامپیوتر میزبان را با کانتینرها به اشتراک می گذارند.تمام منابع سیستم عامل کامپیوتر میزبان بین کانتینرها به اشتراک گذاشته شده و کانتینر ها سخت افزار مجازی و خود سیستم عامل ندارند فقط کانتینرها از کتابخانه ها و فایل های باینری نیاز دارند. [۴]



۳-۱ کانتینرها

۴- پیاده سازی کانتینر با ماشین مجازی

در این روش شما محدود به استفاده از یک زیر ساخت خاص نیستید برنامه های شما به آسانی و سرعت زیاد می توانند از یک زیر ساخت به زیر ساخت دیگر منتقل شوند.

بعضی از فرآیندهایی محل اجرای برنامه مبتنی بر کانتینر شامل

۴-۱ تاخیر: برنامه هایی که به تاخیر حساس هستند بهتر است که روی سرور فیزیکی اجرا شوند.

۴-۲ ظرفیت: ماشین های مجازی جهت استفاده مناسب از ظرفیت بهینه شده اند اگر برنامه مبتنی بر کانتینر شما تمام ظرفیت زیر ساختی فیزیکی رو مسدود نکند مجازی سازی می تواند بهینه باشد.

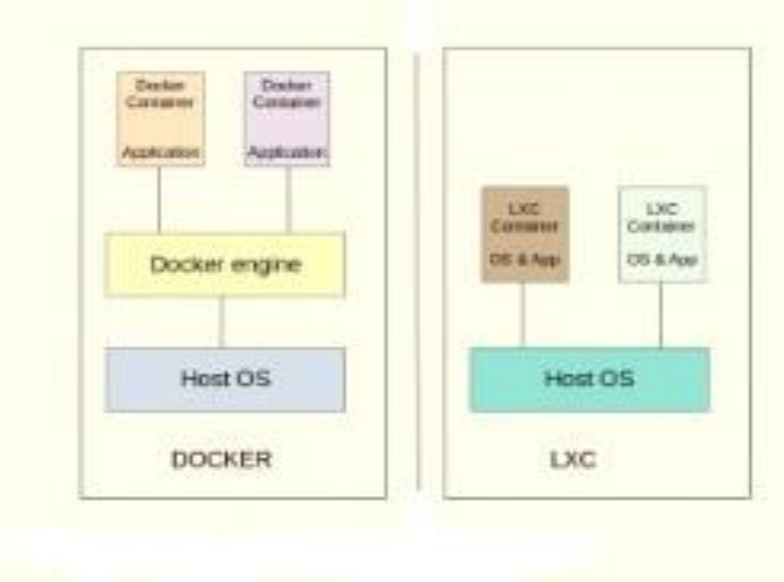
۴-۳ mixed workloads: روی سرور فیزیکی یک سیتم عامل اجرا اگر از ترکیب کانتینر های ویندوزی و لینوکسی استفاده کنید باید از مجازی سازی استفاده کنید.

۴-۴ multi tenancy: کاربران دارای حجم کاری خاصی می باشند و نمی توانند هسته های لینوکسی و ویندوزی را به اشتراک بگذارند در این مورد ماشین های مجازی لایه ای اضافی برای ایزوله سازی علاوه بر کانتینرها ایجاد می کنند.

۴-۵ Resource pools: مجازی سازی از ویژگی هایی برای کنترل اینکه چگونه ماشین مجازی از منابع استفاده کنند را ارائه می- دهند. داکر مفهوم محدودیت منابع را فراهم می کند اما برای سرور فیزیکی مدیریت منابع که به شما بستگی دارد.[۵]

۵-مجازی سازی با کانتینر داکر و LXC

از فضای نام کاربر برای انزوای پایگاه داده کاربر کانتینر و پایگاه داده کاربر میزبان، از فضای نام فرآیند برای تضمین دسترسی و مدیریت کانتینر به تنها فرآیندهای در حال اجرا در آن حامل و از فضای نام شبکه نیز جهت ارایه دستگاه های شبکه و آدرس IP مجازی یک حامل را ستفاده می کند. مدیریت منابع در این سیستم از طریق cgroups ممکن است همچنین کنترل فرآیند نیز توسط cgroups انجام می شود که قابلیت محدود کردن استفاده از پردازنده و انزوای کانتینرها و فرآیندها است. کارایی رویکرد مبتنی بر حامل تقریبا در بسیاری از شرایط برابر با هسته لینوکس است.[۶]



شکل ۵- مجازی سازی با کانتینر داکر و lxc

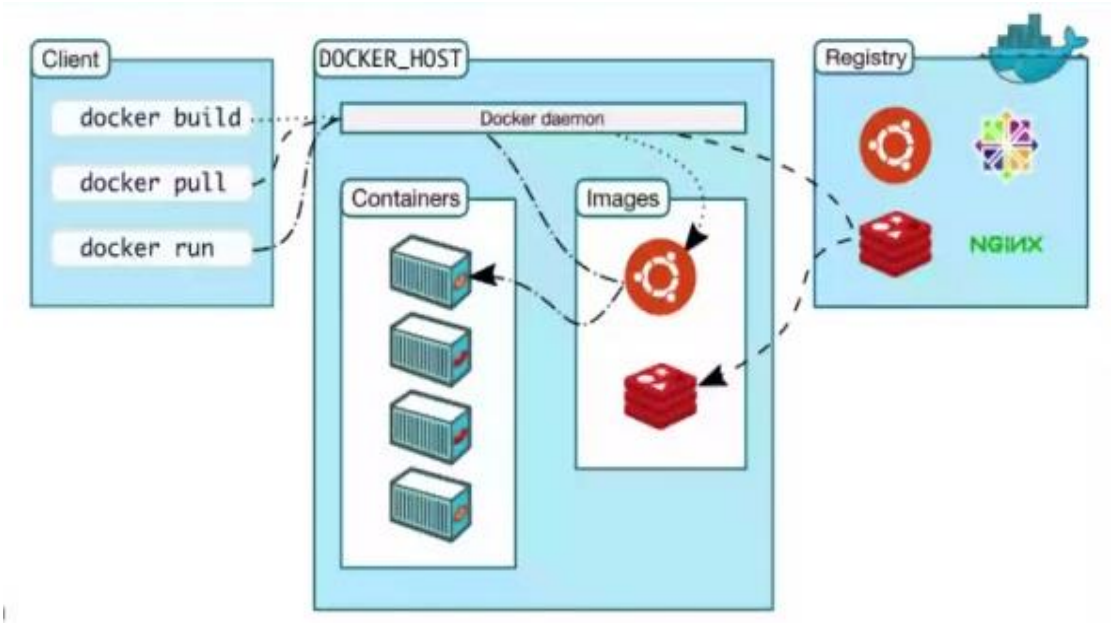
۶-اجزای تشکیل دهنده مجازی سازی با کانتینر داکر

۶-۱ File system : آخرین جزء داکر aufs(advanced multi –layered unification) فایل سیستم است که از آن به عنوان سیستم فایلی برای کانتینرها استفاده می کند وجه تمایز داکر با دیگر ابزارهای متنی بر کانتینر است .aufs سیستم فایلی لایه ای است که می تواند به طور آشکارا یک یا چند سیستم فایل موجود را پوشش دهد یا روی هم قرار دهد. قادر است چندین لایه را به نمایش واحدی از یک سیستم عامل ادغام نماید. در نتیجه در مصرف حافظه و دیسک صرفه جویی و سرعت راه اندازی داکر را افزایش دهند.

۶-۲ Resource management : ابزارهایی را به منظور سهولت ایجاد و کارکردن با کانتینرها را فراهم می کند LXC هسته داکر را تشکیل می دهد ، LXC برای انزوای حامل ها از میزبان از فضاهای نام سطح هسته استفاده می کند. جزء دیگر داکر که توسط LXC ارایه می شود cgroups است که مسئول مدیریت و محدود سازی استفاده از منابع در حامل ها است.cgroups معیارهای گوناگونی از منابعی که مدیریت می کند چاپ می کند و داکر نیز از این معیارها برای نظارت بر مصرف منابع فرآیندهای مختلف کانتینرها استفاده می کند.[۷]

۶-۳ Docker repository یک مزیت بزرگی که مجازی سازی با کانتینر داکر ارائه می دهد رجیستری آن است. بزرگترین رجیستری مربوط به docker hub که انواع ایمیج ها را داخل آن ایجاد و جست وجو کرد.یکی از ویژگی های داکر قابلیت

جست و جو بارگیری راه اندازی ایمیج های کانتینرها است که توسط دیگر توسعه دهندگان ایجاد شده اند مکانی که تصاویر ذخیره می شوند رجیستری نام دارد و داکر یک رجیستری عمومی به نام docker hub ارائه می کند. رجیستری داکر سرویس یا برنامه ای سمت سرویس دهنده است که نوسعه دهندگان می توانند توسط آن تصاویر خود را در مخزن عمومی ذخیره یا push نمایند و می تواند حتی به اشتراک گذارند و دیگر توسعه دهندگان آن را بارگیری یا pull کنند مانند github رجیستری عمومی داکر شامل تصاویری از نرم افزار های آماده اجرا از قبیل پایگاه داده سیستم های مدیریت محتوا محیط های توسعه سرویس دهنده های وب هستند بارگذاری یا بارگیری تصاویر نیازمند ایجاد حسای کاربری در داکر است.[۸]



۳-۶ رجیستری مربوط به docker hub

۷-معماری داکر

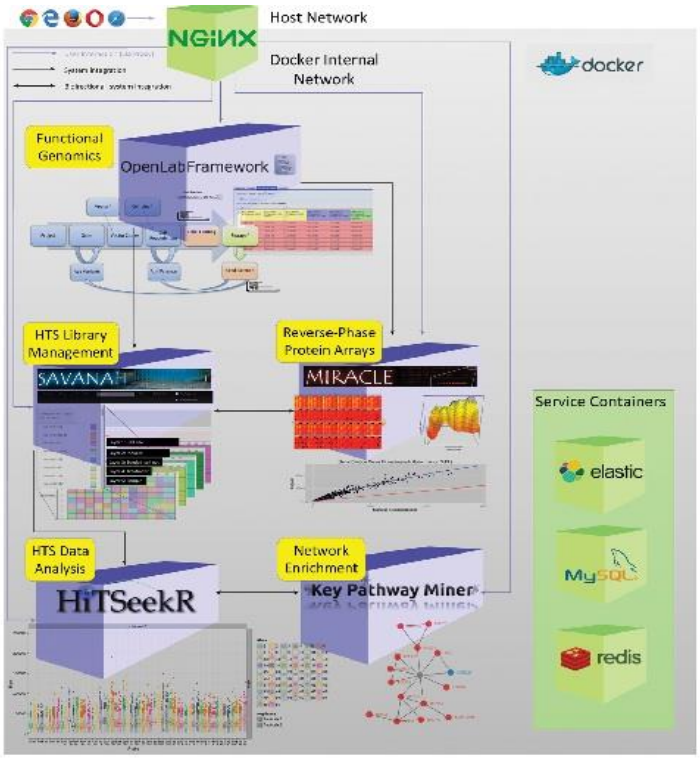
قبل از محیط تنظیم کشف مقصد پلتفرم ما تصویرهای داکر را برای هر یک از ابزارهای نرم افزاری ساختیم. و اضافه به docker hub که آماده سازند آن ها را به آسانی و قابل دسترسی باشند(جدول۱). علاوه بر این، که تضمین کنند که این تصویرها جدیدترین باقی بمانند، ما پیکربندی بکنیم یک گزینه ساخت خودکار شده آن ایجاد شده است. [۹]موقعی که یک commit جدید به متناظر github مخزن متن برنامه ساخته شده است. این توسط اضافه کردن یک docker file با دستور العمل ساخت مهم به ریشه github مخزن به نتیجه رسیده می شود.(https://docs.docker.com/dockerhub/github/) بعد ما ایجاد کردیم مخزن github دیگر، برای محیط کشف مقصد پلتفرم شامل config مهم برای هر یک از ابزارها به خوبی در فایل docker compose.yml با دستور ساخت مشترک که بایگانی بکند.(https://github.com/NanoCAN/Docker-HTS-platform) شکل ۲ تصور که چه طور نمونه سازی کانتینر در برگیرنده داکر در مدت شبکه مجازی منزوی اطلاعات را انتقال دهد و چه طور کانتینر در برگیرنده های یک key-value (MySQL, https://hub.docker.com/_/mysql/) و یک موتور جست و جو (https://(hub.docker.com/_/redis/)redis) تمام دسترسی کاربران browser- based با درخواست ها توسط یک کانتینر در برگیرنده سرویس دهنده وب اختصاصی است کانال یابی شده است. [۱۰]آن چنان که اطلاعات را در نقطه ورود به

پلتفرم را سرویس می نماید. به استثناء کانتینر در برگیرنده های سرویس اما شامل ابزارهای نرم افزاری علمی متن برنامه استفاده کرد اینجا GPLV3 مجوز داده شده است و آزادانه و تعدیل شده تنظیم شده می- شود.[۱۱]

جدول ۷- docker hub urls برای کاربردها وب در محیط کشف پلتفرم استفاده کرد.

OpenLabFramework	https://hub.docker.com/r/nanocan/openlabframe-work/
SAVANAH	https://hub.docker.com/r/nanocan/savanah/
MIRACLE	https://hub.docker.com/r/nanocan/miracle/
	https://hub.docker.com/r/nanocan/rmiracle/
HiTSeekR	https://hub.docker.com/r/nanocan/hitseekr/
KeyPathwayMiner	https://hub.docker.com/r/baumbachlab/keypathwayminer-web/

در شکل زیر دید کلی از high-throughput محیط محافظت با محفظه در برگیرنده های داکر برای هر برنامه کاربردی (جعبه های ارغوانی) اتصال های شبکه pre-defined داخلی محافظت شده ارتباطی بین ابزار تسهیل می کنند جائیکه(فلش های سیاه) نیاز داشتند، محفظه در برگیرنده های خدمات (جعبه های سبز) توسط تمام کاربردهای مشترک شده است. (فلش های سیاه اینجا برای درجه وضوح حذف کردند).یک محفظه در برگیرنده nginc docker وقتی که نقطه ورود را به محیط سرویس می نماید و دستیابی کاربران را فعال کردند توسط ابزارهای مرور گر وب(فلش های ارغوانی رنگ).[۱۲]



شکل ۷- دید کلی از high-throughput محیط محافظت با محفظه در برگیرنده های داکر

۸-مزایای داکر نسبت به ماشین های مجازی

حجم کانتینرها خیلی کمتر از ماشین های مجازی است یک سرور به سادگی قادر خواهد بود که تعداد کانتینر به مراتب بیشتری نسبت به ماشین های مجازی را روی خود مدیریت کند به طوریکه حجم هر ماشین مجازی چند گیگابایت است و اکثر کانتینرها حجم آن ها به چند مگابایت می رسد.

ماشین های مجازی زمان نسبتا قابل توجهی برای بوت شدن سیستم عاملشان نیاز دارند و این در حالی است که کانتینرها خیلی سریعتر اجرا می شوند و صرفا یک پراسس روی سیستم عاملی است که در حال اجرا است.

به جای اجرای یک اپلیکیشن حجیم و پیچیده در قالب یک کانتینر می توان اپلیکیشن خود را به صورت یکسری ماژول مجزا و ناوابسته به یکدیگر طراحی و به صورت یک پکیج کانتینر روی سرور قرار داد.

در کانتینرها مدیریت منابع و ریسورس ها به صورت قدرتمند و متغیر نسبت به ماشین مجازی است

در کانتینرها انتقال پذیری راحت و سبک است.[۱۳][۱٤]

۹- معایبی که در داکر می توان به آن اشاره کرد

یکی از معایب آن اینست که وابسته به یک سیستم عامل است کانتینری که مبتنی بر پلتفرم لینوکس طراحی شده قابل اجرا روی پلتفرم مثلا سیستم عامل دیگری نیست.مورد دیگر که می توان به آن اشاره نمود این است که تهدیدات امنیتی نسبت به مجازی- سازی را در سیستم توزیع می کند به همین دلیل کانتینرها مستقل از سیستم عامل نیستند و چندین کانتینر از یک سیستم عامل واحد استفاده می کنند.[۱۵]

۱۰-امنیت در داکر

داکر در محیط های اشتراکی امنیت را برای نرم افزار ها بوجود می آورد اما به طور کامل امنیت را برقرار نمی کند برنامه نویس و افرادی که با شبکه کار می کنند باید سیستم عامل داکر را جدا امن کنند. شرکت ها از داکر به عنوان تولید نرم فزار های خود و افزودن ویژگی های جدید سرعت با امنیت بالاتر برای تمامی سیستم های لینوکسی و ویندوزی استفاده می کنند.hypervisor امکانات بسیار کم تری نسبت به هسته لینوکس فراهم می کند به همین دلیل کم تر در معرض خطر قرار دارد هسته شامل فایل سیستم شبکه سازی کنترل پردازش است.[۱۶] داکر در حال حاضر یک زیر ساختی دارند که به ادمین ها اجازه می دهند کانتینر را امضاء کنند و با این کار از گسترش کانتینر نامعتبر جلوگیری می کنند و از طرفی دیگر ممکن است آسیب پذیری ها در برخی از نرم افزارها بوجود آید برای اینکه از آسیب پذیری ها مطلع شوند، نرم افزار های مختلفی را برای امنیت وجود دارد که یکی از نرم افزار ها به نام twist lock باعث می شود نرم افزار رفتارهای قابل انتظار پردازش و همچنین فعالیت های شبکه مانند IPپورت مقصد و مبدا را در قالب پروفایل انجام دهد و در نتیجه رفتار غیر منتظره را شناسایی کند. یکی دیگر از راه های امنیت در کانتینر استفاده از کمپانی polyverse به این شکل هست که اپلیکیشن در زمان کوتاه می تواند شروع به کار کند و این باعث می شود که هکر فرصت زیادی برای اجرای اپلیکیشن خود در کانتینر نداشته باشد.[۱۷]

۱۱-راه کار پیشنهادی داکر برای بهینه سازی سیستم عامل و مدیریت کانتینرها

یک راه‌کار کاملا خودکار برای نگهداری و راه‌اندازی کانتیرها استفاده از Community Edition در داکر است. تمام آن چیزی که شما به آن نیاز دارید و خودتون باید ایجاد کنید را به صورت یک‌پارچه در اختیار شما قرار می‌دهد. [۱۸] [۱۹]

۱۱-۱ بر روی سیستم‌عامل و پلتفرم‌های مختلف در دسترس خواهد بود.

۱۱-۲ دارای مانیتورینگ و احراز هویت دیجیتال به همراه اسکن امنیتی کانتینرها می‌باشد.

۱۱-۳ به خوبی با اپلیکیشن‌های مختلف یکپارچه شده و می‌تواند فرآیندهای مختلف را به صورت خودکار راه‌اندازی کرد.

۱۱-۴ همواره دارای آخرین نسخه‌ی پایدار داکر می‌باشد.

۱۱-۵ دارای پشتیبانی به همراه SLA است.

۱۱-۶ دارای قابلیت پشتیبانی از نسخ مختلف و قدیمی و ارائه‌ی پچ‌های امنیتی و به روزرسانی‌های لازم می‌باشد.

۱۱-۷مجموعه کاملی از خدمات و آموزش‌ها برای سرعت بخشیدن به سرویس‌دهی است.

۱۱-۸برای استفاده از آن نیاز است تا ثبت نام کرده و هزینه پرداخت کرد.

نتیجه گیری

داکر به خاطر کانتینر از اهمیت ویژه ای برخوردار است و یک فناوری محسوب می شود که می تواند مشکلات انتقال یک نرم افزار را از یک سیستم به سیستم دیگر را به راحتی انجام دهد کانتینر ها به توسعه دهندگان کمک می کند تا اپلیکیشن های خود را راحت و سریع با حجم کم تری و به صورت قابل حمل در یک کانتینر قرار دهند که این اپلیکیشن به صورت مجازی قابل اجرا هستند همچنین می توان داکر را به عنوان یک گیت هاب در نظر گرفت مخزن گیت هاب کمک می کند تا نرم افزار بهینه تر و اجرا و مدیریت را ارتقاء دهد. [۲۰]

مراجع

[1] Yu Jin-Gang, Zhai Ya-Rong1, Yu Bo, Li Shu.(2017). Research and Application of Auto-scaling Unified Communication Server Based978-1-5386-1230-9/17 $31.00 © 2017 IEEE-
DOI 10.1109/ICICTA.2017.41

[2] Olivier Sallou, Cyril Monjeaud.(2015) GO-Docker. A batch scheduling system with Docker-containers. 978-1-4673-6598-7/15 $31.00 © 2015 IEEEDOI 10.1109/CLUSTER.2015.89

[۳] Olivier Sallou, Cyril Monjeaud.(2015) GO-Docker. A batch scheduling system with Docker -containers. 978-1-4673-6598-7/15 $31.00 © 2015 IEEEDOI 10.1109/CLUSTER.2015.89

[۴] Olivier Sallou, Cyril Monjeaud.(2015) GO-Docker. A batch scheduling system with Docker-containers. 978-1-4673-6598-7/15 $31.00 © 2015 IEEEDOI 10.1109/CLUSTER.2015.89

[5] alin.calinciuc, cristian.spoiala, turcu, filote.(2016). OpenStack and Docker: building a high-performance IaaS platform for interactive social media applications. 978-1-5090-1993-9/16/$31.0 - ©2016 IEEE.

[6] Manu A R, Jitendra Kumar Patel, Shakil Akhtar, V K Agrawal PhD, K N Bala Subramanya-Murthy.(2016) Docker Container Security via Heuristics-BasedMultilateral Security- Conceptual- and Pragmatic. 978-1-5090-1277-0/16/$31.00 ©2016 IEEE.

[7] Olivier Sallou, Cyril Monjeaud.(2015) GO-Docker. A batch scheduling system with Docker-containers. 978-1-4673-6598-7/15 $31.00 © 2015 IEEEDOI 10.1109/CLUSTER.2015.89

[8] Abdulrahman Azab.(2017). Enabling Docker Containers for High-Performance and Many-Task-Computing. 978-1-5090-5817-4/17 $31.00 © 2017 IEEEDOI 10.1109/IC2E.2017.52

[9] Olivier Sallou, Cyril Monjeaud.(2015) GO-Docker. A batch scheduling system with Docker-containers. 978-1-4673-6598-7/15 $31.00 © 2015 IEEEDOI 10.1109/CLUSTER.2015.89

[10] Kitti Klinbua, Wiwat Vatanawood.(2017).Translating TOSCA into Docker-ComposeYAML File-Using ANTLR 978-1-5386-0497-7/17/$31.00 ©2017 IEEE

[11] markus list.(2017). Using Docker Compose for the SimpleDeployment of an Integrated Drug - TargetScreening Platform. DOI: 10.1515/jib-2017-0016

[12] markus list.(2017). Using Docker Compose for the SimpleDeployment of an Integrated Drug - TargetScreening Platform. DOI: 10.1515/jib-2017-0016

[13]] alin.calinciuc, cristian.spoiala, turcu, filote.(2016). OpenStack and Docker: building a high-performance IaaS platform for interactive social media applications. 978-1-5090-1993-9/16/$31.0 - ©2016 IEEE

[14] Suchit Dhakate, Prof. Anand Godbole.(2015). Distributed Cloud Monitoring Using Docker as-NextGeneration Container Virtualization Technology. 978-1-4673-6540-6/15/$31.00 ©2015 IEEE.

[15] Nitin Naik.(2016). Building A Virtual System of Systems Using DockerSwarm in Multiple-Clouds. 978-1-5090-0793-6/16/$31.00 ©2016 Crown.

[16] Abdulrahman Azab.(2017). Enabling Docker Containers for High-Performance and Many-Task-Computing. 978-1-5090-5817-4/17 $31.00 © 2017 IEEE .DOI 10.1109/IC2E.2017.52.

[17] Abdulrahman Azab.(2017). Enabling Docker Containers for High-Performance and Many-Task - Computing. 978-1-5090-5817-4/17 $31.00 © 2017 IEEE .DOI 10.1109/IC2E.2017.52.

[18] alin.calinciuc, cristian.spoiala, turcu, filote.(2016). OpenStack and Docker: building a high-performance IaaS platform for interactive social media applications. 978-1-5090-1993-9/16/$31.0 - ©2016 IEEE.

[19] Jeeva Chelladhurai, Pethuru Raj Chelliah, Sathish Alampalayam Kumar.(2016). Securing-Docker Containers from Denial of Service (DoS) Attacks. 978-1-5090-2628-9/16 $31.00 © 2016 IEEE.

[20] Walter Blair, Aspen Olmsted, Paul Anderson.(2017). Docker vs. KVM: Apache Spark – application performance and ease of use. 978-1-908320/93/3/$31.00©2017 IEEE.