

دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

دانشكده‌ی مهندسی کامپیوتر

تشخیص نفوذ شبکه‌های کامپیوتری

مبتنی بر یادگیری‌ماشین

نگارش

بهار کاویانی

استاد راهنما

دکتر رضا صفابخش

اردیبهشت‌ماه ۱۴۰۰



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

دانشكده‌ی مهندسی کامپیوتر

گزارش نوشتاری موضوع

تشخیص نفوذ شبکه‌های کامپیوتری

مبتنی بر یادگیری‌ماشین

نگارش

بهار کاویانی

استاد راهنما

دکتر رضا صفابخش

اردیبهشت‌ماه ۱۴۰۰

سپاس‌گزاری

اینجانب بهار کاویانی مراتب تقدیر و تشکر خود را نسبت به استاد راهنمای خود، آقای دکتر رضا صفابخش که طی تدوین این گزارش نوشتاری همواره مرا یاری نموده‌اند، ابراز می‌دارم.

بهار کاویانی

اردیبهشت‌ماه ۱۴۰۰

چكیده

در رایانش ابری به واسطه وجود مجازی سازی این امکان به وجود آمده است که یک سیستم بتواند محیط‌های مختلفی را برای مشتری‌‌ها مختلف و یا یک مشتری فراهم آورد. برای مجازی سازی دو راهکار وجود دارد؛ راهکار نخست که همان راهکار معمول است مجازی سازی عادی یا hypervisor و راهکار دیگر مجازی سازی بر اساس کانتینر است. در این مقاله به مجازی سازی برپایه کانتینر پرداخته میشود.

این مقاله تلاش میکند تا یه مرور جامع از فناوری‌های کانتینر برای مجازی سازی و پویایی ناشی از آن داشته باشد. به واسطه این مرور افراد میتوانند با توجه به نیاز خود از این تکنولوژی استفاده کنند.

این مقاله نیز چون به بیان علت‌های استفاده از کانتینربندی اشاره میکند مقاله مناسبی هست. همچنین این مقاله ابزار‌های مفید برای این کار را نیز معرفی میکند.

واژه‌های کلیدی:

ماشین مجازی، کانتینر، داکر، مجازی سازی،‌هایپروایزر

|  |  |
| --- | --- |
| فهرست مطالب | صفحه |

[فصل اول مقدمهمقدمه 1](#_Toc45998994)

[فصل دوم ماشین مجازیماشین مجازی 4](#_Toc45998995)

[2-1- مزایای استفاده از ماشین مجازی 6](#_Toc45998996)

[2-2- معایب استفاده از ماشین مجازی 7](#_Toc45998997)

[2-3- کاربردهای ماشین مجازی 8](#_Toc45998998)

[2-4- ابزارهای ماشین مجازی 9](#_Toc45998999)

[فصل سوم کانتینربندی کانتینربندی 11](#_Toc45999000)

[3-1- مزایای استفاده از کانتینربندی 13](#_Toc45999001)

[3-1-1- حجم تصویر 13](#_Toc45999002)

[3-1-2- سرعت 13](#_Toc45999003)

[3-1-3- سرعت دیپلوی 13](#_Toc45999004)

[3-1-4- میکروسرویس ها 14](#_Toc45999005)

[3-1-5- نسخه‌بندی 14](#_Toc45999006)

[3-2- معایب استفاده از کانتینربندی 14](#_Toc45999007)

[3-2-1- امنیت 14](#_Toc45999008)

[3-2-2- نیازمندی‌های سیستم 15](#_Toc45999009)

[3-2-3- مدیریت 15](#_Toc45999010)

[3-3- داکر 16](#_Toc45999011)

[3-3-1- کاربرد 16](#_Toc45999012)

[3-3-2- مقیاس پذیری داکر 17](#_Toc45999013)

[3-3-3- قابل حمل بودن داکر 17](#_Toc45999014)

[3-3-4- حجم کم تصاویر در داکر 17](#_Toc45999015)

[3-4- خلاصه 18](#_Toc45999016)

[فصل چهارم مقایسه و بررسی ماشین مجازی و سیستم‌‌های مبتنی بر کانتینرمقایسه و بررسی ماشین مجازی و سیستم‌‌های مبتنی بر کانتینر 19](#_Toc45999017)

[4-1- انزوا 20](#_Toc45999018)

[4-2- سیستم عامل 21](#_Toc45999019)

[4-3- عملکرد 21](#_Toc45999020)

[4-4- حجم تصاویر 21](#_Toc45999021)

[4-5- زمان راه اندازی 21](#_Toc45999022)

[فصل پنجم جمع‌بندی و نتیجه‌گیری و پیشنهاداتجمع‌بندی و نتیجه‌گیری 23](#_Toc45999023)

[منابع و مراجع 25](#_Toc45999024)

|  |  |
| --- | --- |
| فهرست اشكال و جداول | صفحه |

[شکل 1-1 دیاگرام سیستم‌‌های مبتنی بر ماشین مجازی 3](#_Toc45998884)

[شکل 1-2 دیاگرام سیستم‌‌های مبتنی بر کانتینر 3](#_Toc45998885)

[جدول 2-1 اطلاعات نرم افزارهای تولید و مدیریت ماشین مجازی 10](#_Toc45998886)

[شكل ۳-1 لوگوی داکر 12](#_Toc45998887)

[جدول 4-1 خلاصه مقایسه ماشین مجازی و کانتینر 22](#_Toc45998888)

# فصل اول مقدمهمقدمه

هر انسانی که تجربه برنامه نویسی شبکه را در این دوران داشته باشد احتمالا تا به حال کلمه داکر به گوشش خورده است. وکسانی که دستی بر آتش دارند آن را به زبان ساده به این صورت بیان میکنند که داکر و یا کلا سیستم‌‌ها مبتنی بر کانتینر یک ابزار کارآمد برای بسته بندی کردن، جابه جا کردن و اجرا برنامه‌‌ها است. قبل از این که بخواهیم به خود تعریف داکر برسیم میبایست با مفاهیم اصلی کانتینر و ماشین مجازی آشنا شویم .

کانتینر‌‌ها و ماشین‌های مجازی هدفی یکسان دارند؛ جدا سازی یک برنامه (Application) و نیازمندی‌هایش در قالب بسته بندی شده که بتواند در هر محیطی اجرا شود. اگر بخواهیم دقیق تر بگوییم این دو مفهوم با ایجاد یه سخت افزار مجازی برای نرم افزارها تصوری یکسان از سخت افزار ایجاد میکنند. اما تفاوت این دو سیستم در معماری آن هاست که در ادامه ذکر خواهد شد.

ماشین مجازی در واقع یک شبیه‌سازی از کامپیوتر واقعی است؛ که برنامه حس میکند در حال استفاده از یک کامپیوتر واقعیست.

ماشین‌های مجازی رای یک کامپیوتر فیزیکی توسط ‌هایپروایز اجرا میشوند.‌هایپروایز یک مدل از مجازی سازی سخت افزار(hardware virtualization) است که به شما امکان اجرای چندین سیستم عامل را در یک زمان روی یک سیستم میزبان را فراهم میکند. که در این حالت سیستم عامل‌های مجازی نصب شده همانند سیستم عامل واقعی امکان استفاده از منابع سخت افزاری موجود درسیستم مانند پردازنده، حافظه جانبی و حافظه‌ی را دارد. ‌هایپروایزر وظیفه برطرف کردن نیازمندی‌های سیستم عامل‌های مهمان، مدیریت ارتباط بین آنها و میزان استفاده ی آن‌‌ها از منابع سخت افزاری را برعهده دارد. ازهایپراایز با عنوان دیگری هم نام برده میشود که مدیر ماشین مجازی نام دارد. هرسیستم عامل مهمانی که در این مدیر ماشین مجازی اجرا میشود دارای برنامه‌‌ها و نیازمندی‌های خاص خود است که به صورت کامل وجود دارند به همین دلیل است که میگوییم ماشین مجازی از نظر برنامه مهمان(سیستم عامل) مانند یک ماشین کامل است. ماشین مجازی vmware Workstation 8 یکی از‌هایپروایزرهای مشهور است که احتمالا هر برنامه نویسی با آن کار کرده است.

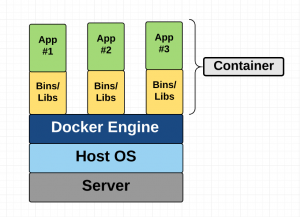
تمام اونچه توضح داده شد رو در عکس بالا میبینید نکته دیگه اینکه یک مدل دیگه مجازی سازی هم داریم که در مقاله جداگانه ای در موردش صحبت خواهد شد که مقداری با این مدل فرق داره. که در واقع اون مدل در سطح production استفاده بشه

در شکل زیر جایگاه‌هایپروایزر در سیستم را میبینیم.

شکل 1-1 دیاگرام سیستم‌‌های مبتنی بر ماشین مجازی

بر خلاف ماشین مجازی‌‌ها که مجازی سازی را در لایه سخت افزار انجام میدهند؛ سیستم‌های برمبنای کانتینر، مجازی سازی را در لایه سیستم عامل انجام میدهند.

همونطور که گفته شد کانتینر‌‌ها و ماشین مجازی‌‌ها برای یک هدف طراحی شده اند، و آن جدا کردن فضا و ایزوله سازی نرم افزارهای مهمان از ماشین اصلی است. اما تفاوت بزرگ سیستم‌های مبتنی بر کانتینر و ماشین مجازی این است که کامپیوتر میزبان محیط هسته را با کانتینرها به اشتراک میگذارد ولی در سیستم‌های‌هایپروایزری این اتفاق رخ نمیدهد

در تصویر زیر ساختار سیستم‌های مبتنی بر کانتینر به تصویر کشیده شده است.

شکل 1-2 دیاگرام سیستم‌‌های مبتنی بر کانتینر

# فصل دوم ماشین مجازی**ماشین مجازی**

ماشین مجازی نوعی نرم افزار است که به عنوان یک رایانه مجازی عمل می‌کند. این برنامه بر روی سیستم‌عامل فعلی شما (که سیستم میزبان نام دارد) اجرا می‌شود؛ ماشین مجازی در واقع یک کامپیوتر است که بر روی سیستم عامل شما به صورت جداگانه اجرا می‌شود و شما می‌توانید برای آن از سخت افزار اصلی خودتان سخت افزار مجازی تعیین کنید و آن را هر زمان که میل داشتید ارتقاء بدهید.سیستم‌عامل میهمان دقیقاً مانند هر برنامه دیگری در یک پنجره بر روی سیستم‌عامل میزبان اجرا می‌شود.از منظر سیستم‌عامل میهمان، ماشین مجازی یک رایانه فیزیکی واقعی است.

چون این نرم افزار یک کامپیوتر مجازی را برای شما ایجاد میکند به همین خاطر به آن ماشین مجازی میگویند. و به نرم‌افزاری که امکان ایجاد و اجرای یک ماشین مجازی را فراهم می‌کند‌هایپروایزرگفته می‌شود.

سیستم فیزیکی که ماشین مجازی روی آن اجرا می‌شود ماشین میزبان (Host Machine) نامیده می‌شود و هر ماشین مجازی را نیز یک ماشین مهمان (Guest Machine) می‌نامند.

ماشین مجازی سخت‌افزارهای مجازی خود را دارد که شامل پردازنده، حافظه، هارددیسک، رابط شبکه و دیگر دستگاه‌ها است. این دستگاه‌های سخت‌افزار مجازی از طرف ماشین مجازی به سخت‌افزارهای واقعی روی رایانه نگاشت می‌شوند.

شما می‌توانید چندین ماشین مجازی روی سیستم خود نصب کنید. زمانی که چند سیستم‌عامل روی ماشین مجازی خود نصب کنید، می‌توانید آن را باز کرده و به راحتی سیستم‌عاملی که می‌خواهید بوت(عمل وارد شدن به سیستم عامل کامپیوتر) کنید را انتخاب نمایید. ماشین مجازی سیستم عامل جدید را مانند یک نرم افزار در یک پنجره جدید در سیستم عامل شما باز می‌کند و دقیقا تمام امکانات آن سیستم عامل را با تمام جزئیات در اختیار شما قرار می دهند.

## مزایای استفاده از ماشین مجازی

با این که در عصر کنونی استفاده از کانتینربندی بسیار پررونق شده، اما با این حال دلایلی هست که هنوز هم استفاده از ماشین مجازی بر استفاده از فناوری‌های مبتنی بر کانتینر مانند داکر و غیره ارجحیت داشته باشد.

* + 1. **انعطاف پذیری**

شاید بزرگ‌ترین نقطه قوت برای ماشین‌های مجازی این است که در نهایت، آن‌ها همچنان بیش‌ترین مقدار انعطاف‌پذیری را ارائه می‌دهند. یک ماشین مجازی می‌تواند تقریبا در هر جایی، صرف‌نظر از سیستم عامل و یا پیکربندی میزبان آن مستقر شود. یک سیستم ویندوز می‌تواند میزبان یک ماشین مجازی مبتنی بر لینوکس باشد و بالعکس.

انعطاف‌پذیری

اگرچه کانتینرها مقداری انعطاف‌پذیری خوبی ارائه می‌کند. یک برنامه لینوکس کانتینربندی شده به آن اهمیتی نمی‌دهد که چه نسخه ای از لینوکس برروی میزبان آن باشد. اما شما هنوز نمی‌توانید یک کانتینر لینوکس را بر ویندوز و یا یک کانتینر ویندوز را در لینوکس اجرا کنید، مگر این که از یک ماشین مجازی برای این کار ایجاد کنید.

* + 1. **امنیت و انزوا**

امنیت کانتینر از زمان به وجود آمدن داکر در سال ۲۰۱۳ به شدت بهبود یافته‌است. اما هنوز برخی دلایل نگرانی را در بر دارد؛ در واقع، نگرانی‌های امنیتی دلیل اصلی این هستند که چرا برخی از تیم‌ها استفاده از کانتینرها را انتخاب نمی‌کنند.

برخی از این نگرانی‌ها ممکن است تسکین داده شوند چون پلتفرم‌های کانتینرها به بلوغ می‌رسند و ابزارهای امنیتی بیشتری به طور کامل از آن‌ها پشتیبانی می‌کنند. با این حال، بعید است که کانتینرها به طور کامل با ماشین‌های مجازی از نظر امنیتی مشابه هم باشند، به خاطر این واقعیت ساده که برنامه‌های مبتنی بر کانتنیر هرگز نمی‌تواند به اندازه ماشین‌های مجازی از هم دیگر جدا شوند. ماشین‌های مجازی، هسته یا همان کرنل و یا سایر منابع ضروری سیستم را در برخلاف کانتینرها با هم تقسیم نمی‌کنند.

* + 1. **مدیریت**

درست است که کانتینرها از بسیاری جهات برای محیط‌های میزبان توزیع‌شده مناسب‌تر هستند، این ویژگی نیز می‌تواند مدیریت آن‌ها را سخت‌تر کند. زمانی که صدها کانتینر در بسیاری از سرورها پخش می‌شوند، کنترل اوضاع به سرعت دشوار می‌شود. به همین دلیل است که از ابزارهای سازمان دهنده‌ای مثل Kubernetes استفاده میشود که بخش اعظم پیچیدگی ناشی از زیاد شدن کانتینرها در سیستم‌های توزیع شده را مدیریت می‌کند. با این حال، ابزار سازمان دهنده خودش یک لایه دیگر از پیچیدگی را اضافه می‌کند که شما باید راه‌اندازی، مدیریت و امنیت آن را تامین کنید.

وقتی از ماشین مجازی استفاده میشود به واسطه کمتر شدن امکان پویایی مدیریت آسانتر میشود و همپوشانی کمتری در لایه‌های زیر ساخت وجود دارد.

## معایب استفاده از ماشین مجازی

. با توجه به مزایای گفته شده در قسمت قبل اما روز به روز شاهد پیشرفت پلتفرم‌‌ها مبتنی برا کانتینر هستیم و تقاضای موجود برای پیاده سازی با کانتینر از هایپروایزر به طور روزافزون در حال افزایش است؛ برای بررسی این موضوع به برخی از معایب استفاده از ماشین مجازی میپردازیم.

* + 1. **سرعت**

در برخی جنبه‌ها، ماشین‌های مجازی کندتر هستند. آن‌ها بیشتر طول می‌کشند (این اختلاف گاها به این صورت است که اگر یک برنامه در یک کانتینر در چند ثانیه اجرا شود ممکن است در ماشین مجازی در حد یکی دو دقیقه طول بکشد) از طرفی ممکن است به خاطر محدودیت‌های گذاشته شده برا روی هر ماشین مجازی امکان اجرای برخی از برنامه‌‌ها وجود نداشته باشد.

* + 1. **تکرار**

ماشین‌های مجازی با این فرض طراحی شدند که هر ماشین بر روی یک سرور واحد قرار می‌گیرد. در حالی که امکان ایجاد افزونگی برای ماشین‌های مجازی با پخش آن‌ها در میان یک خوشه از سرورها، باعث شده است که حجم زیادی از داده در بین سرورهای یک خوشه تکرار شود در حالی که این ایراد در کانتینرها وجود ندارد.

* + 1. **تصاویر حجیم**

یک تصویر(Image) ماشین مجازی که شامل یک سیستم‌عامل میزبان (همان طور که بیشتر انجام می‌دهند) معمولا حداقل چند گیگابایت فضا و احتمالا بسیار بیشتر را مصرف می‌کند. در مقایسه، تصویر یک کانتینر می‌تواند تنها چند مگابایت باشد، چون تصویر یک کانتینر نیاز به بسته بندی یک سیستم‌عامل به صورت کامل ندارد. برای مثال

## کاربردهای ماشین مجازی

شاید برای شما پیش آمده باشد که دلتان هوای بازی با برنامه ای را که درکودکی با آن سرگرم بوده و با دوستان خود با آن خاطره داشته اید را کرده باشد.در این صورت ممکن است آن نرم افزار قابلیت نصب روی سیستم عامل فعلی شما را نداشته باشد و مثلا فقط روی ویندوزxp اجرا شود در این صورت شما میتوانید با استفاده از یک ماشین مجازی ویندوز xpرا روی سیستم عامل فعلیتان نصب کنید تا بتوانید آن بازی را روی سیستم عامل ماشین مجازی خود نصب و استفاده کنید. اما کاربردهای ماشین مجازی به همین کاربردهای روزمره محدود نمیشوند در این بخش میخواهیم سه مورد از اصلی ترین کاربردهای ماشین مجازی را معرفی کنیم.

* + 1. **چند سیستمی**

شرکت هایی که به فروش سرور میپردازند عمدتا از این روش استفاده میکنند که یک کامپیوتر بزرگ را بین چندین مشتری تقسیم میکنند، اصولا در چنین روش هایی، شرکت فروشنده با استفاده از تکنولوژی ماشین مجازی برای هر کاربر این امکان را به وجود می‌آورد که حس استفاده از یک کامپیوتر کامل را تجربه کند.

* + 1. **تست برنامه**

ماشین‌‌های مجازی این امکان را به شما میدهند که چنانچه برای تولید و توسعه یک نرم افزار نیاز به ایجاد یک بستر داشته باشید، بتوانید این بستر را از ماشین خود ایزوله کنید و آن را در محیط ماشین مجازی اجرا کنید. در این حالت شما میتوانید بعد از اجرا و تست برنامه خود حتی تصویری از ماشین مجازی تهیه کنید و آن را به دست مشتری برسانید.

* + 1. **استفاده از سیستم عامل دیگر**

گاهی رخ میدهد که کاربر بخواهد برای مدت محدودی از یک سیستم عامل دیگر استفاده کند. در این شرایط به جای این که شخص این سیستم عامل را روی ماشین خود نصب کند میتواند با استفاده از ماشین مجازی از آن استفاده کند.

## ابزارهای ماشین مجازی

تابه حال حتمن اسم برنامه vmware workstation را شنیده‌اید. این برنامه یکی از محبوب ترین برنامه‌‌های ایجاد ماشین مجازی است؛ اما طیف وسیعی از این برنامه‌‌ها وجود دارد. در جدول زیر تعدادی از برنامه‌‌های موجود برای ایجاد ماشین مجازی و مدیریت آن‌‌ها را میبینیم.

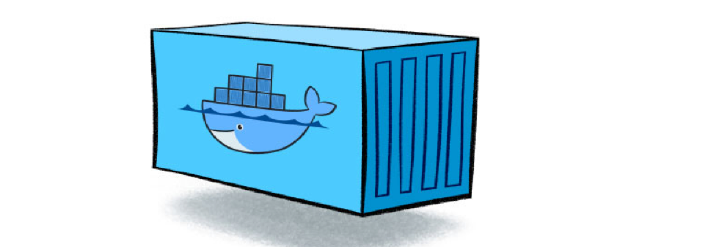
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| نام | سیستم عامل میزبان | سیستم عامل میهمان | ویژگی ها | رایگان بودن |
| [SolarWinds Virtualization Manager](https://try.solarwinds.com/lp/virtualization-manager?CMP=BIZ-RVW-SWTH-vm_mgr-VM-TRY-virtsw) | ویندوز | همه سیستم عامل ها | قابلیت استفاده در فضای ابری | نیست |
| [V2 Cloud](https://v2cloud.com/?referrer=d42d1d2810c5b902e16313e3647f01686XaeD2bHbIO0zEGN4SQmX7UvHx6KqDeh) | ویندوز | ویندوز | امکان دسترسی از طریق مرورگر | نیست |
| [VM Ware Fusion](https://www.vmware.com/products/fusion.html) | مک | ویندوز و لینوکس | نیازی به raboot ندارد | نیست |
| [Parallels Desktop](https://www.parallels.com/products/desktop/) | لینوکس | مک و ویندوز | نیازی به raboot ندارد | نیست |
| [Oracle Virtualization](https://www.oracle.com/virtualization/virtualbox/index.html) | مک، ویندوز، لینوکس | همه سیستم عامل ها | - | است |
| [VM Ware Workstation](https://www.vmware.com/products/workstation-pro.html) | ویندوز و لینوکس | همه سیستم عامل ها | قابلیت کار بر روی فضای ابری دارد | نیست |
| [QEMU](https://www.qemu.org/) | همه سیستم عامل ها | همه سیستم عامل ها | قابلیت استفاده در همه پلتفرم‌‌ها را دارد | است |
| [Virtual PC](https://www.microsoft.com/en-us/download/details.aspx?id=3702) | ویندوز7 | ویندوزxp و vista | - | است |
| [Microsoft Hyper-V](https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v) | ویندوز8 و بالاتر | ویندوز و لینوکس | قابلیت استفاده روی azure را دارد | است |
| [Redhat Virtualization](https://www.redhat.com/en/technologies/virtualization/enterprise-virtualization) | لینوکس | ویندوز و لینوکس | متن باز است | است |
| [Veertu-for MAC](https://veertu.com/anka-technology/) | مک | مک و ios | برای اجرای برنامه‌‌های روی ios مناسب است | نیست |
| [Apple-Boot Camp](https://www.apple.com/) | مک | ویندوز | نیازی به دانلود ندارد | است |

جدول 2-1 اطلاعات نرم افزارهای تولید و مدیریت ماشین مجازی

# فصل سوم کانتینربندی کانتینربندی

برنامه نویس همواره با این مشکل مواجه هستند که مثلاً اپلیکیشنی را روی سیستم خود کدنویسی، دیباگ، تست و اجرا کرده و از صحت عملکرد آن اطمینان حاصل می‌کنند اما همین که آن را به محیط یا پلتفرم دیگری همچون یک سرور دیپلوی می‌کنند، باگ‌ها و مشکلات عجیب‌وغریب بسیاری را تجربه می‌کنند و اینجا است که Container خودنمایی می‌کند. به عبارت دیگر، کانتینر این اطمینان را حاصل می‌کند که نرم افزار فارغ از اینکه روی چه پلتفرمی دیپلوی گردد، کاملاً به درستی اجرا گردد به طوری که روی سیستم دولوپر گرفته تا محیط تست،‌ سرور مجازی یا حتی کلود عملکرد یکسانی خواهد داشت.

حال ممکن است این پرسش پیش آید که چه دلیلی ممکن است وجود داشته باشد که اپلیکیشن پس از دیپلوی در محیط‌های مختلف به درستی اجرا نگردد. در پاسخ به این سوال فرض کنیم اپلیکیشنی نوشته‌ایم که در حین فرایند توسعه، نسخه‌ی 7.1 PHP روی لپ‌تاپ‌مان نصب بوده است و از برخی ویژگی‌های این نسخه همچون اپراتور ?? در کدهای خود استفاده کرده‌ایم. حال زمانی که پروژه را روی سرور اصلی که نسخه PHP 5.6 را دارا است دیپلوی می‌کنیم، مسلماً به مشکل برخورد خواهیم کرد.

 اینجا است که کانتینر به کمک تیم‌های توسعه نرم‌افزار می‌آید به طوری که این فناوری شامل یک محیط اجرای کامل است که اپلیکیشن به انضمام کلیه وابستگی‌هایش من‌جمله کتابخانه ها، فریمورک‌ها، فایل‌های پیکر‌بندی و غیره را شامل می‌شود که در قالب یک پکیج عرضه می‌شوند (در واقع، وجه تسمیه کانتینرها به خاطر شباهتی است که با کانتینرهایی دارند که در تصویر لوگوی داکر مشاهده می‌شوند. به عبارت دیگر، هر کانتینر محیطی کاملاً مجزا است که داخل آن کالاهای مختلفی را می‌توان قرار داد اما این در حالی است که تمامی کانتینرها روی یک پلتفرم واحد، کشتی، قرار دارند.)

شكل ۳-1 لوگوی داکر

## مزایای استفاده از کانتینربندی

باتوجه به اقبال عمومی ایجاد شده در جامعه برنامه نویسان برای استفاده از تکنولوژی مبتنی بر کانتینر به برررسی موارد مثبت در این روش میپردازیم.

### حجم تصویر

با توجه به اینکه حجم کانتینرها خیلی کمتر از ماشین‌های مجازی است، یک سرور به سادگی قادر خواهد بود تعداد کانتینر به مراتب بیشتری نسبت به ماشین مجازی را روی خود میزبانی کند و نیاز به توضیح نست که حجم هر ماشین مجازی ممکن است به چند گیگابایت برسد اما حجم اکثر کانتینرها نهایتاً به چند صد مگابایت خواهد رسید.

### سرعت

نکته دیگری که در ارتباط با مزیت‌های کانتینرها وجود دارد این است که ماشین‌های مجازی زمان نسبتاً قابل‌توجهی برای بوت شدن سیستم‌عامل‌شان نیاز دارند و این در حالی است که کانتینرها خیلی سریع‌تر اجرا می‌شوند چرا که کانتینر صرفاً یک Process روی سیستم‌عاملی می‌باشد که در حال اجرا است.

### سرعت دیپلوی

این امکان فراهم شده تا اصطلاحاً یک ایمِج از کانتینری را به عنوان کانتینر بِیس (پایه) در نظر گرفته و دیگر ایمِج‌ها وابسته به آن باشند. به عبارت دیگر، می‌شود یک ایمِج پایه‌ای که شامل سیستم‌عامل، فایل‌های پیکربندی و دیگر وابستگی‌ها همچون لایبرری‌ها و فریمورک‌ها می‌باشد را ایجاد کرد سپس تیم توسعه نرم‌افزار صرفاً ایمِجی حاوی سورس‌کد اپلیکیشن که مبتنی بر آن ایمِج پایه‌ای است را ایجاد کنند و همین مسئله مجدداً منجر به سرعت توسعه‌ دیپلوی نرم‌افزار خواهد شد.

### میکروسرویس ها

مزیت دیگر استفاده از کانتینرها امکان بهره بردن از اپلیکیشن‌های ماژولار است. به عبارت دیگر، به جای اجرای یک اپلیکیشن حجیم و پیچیده در قالب یک کانتینر، می‌توان اپلیکیشن خود را به صورت یکسری ماژول مجزا که اصطلاحاً تحت عنوان **میکروسرویس** شناخته می‌شوند طراحی کرده و هر یک از آن‌ها را به عنوان یک پکیج روی سرور قرار داد به طور کلی، اپلیکیشن‌هایی که با معماری میکروسرویس طراحی می‌شوند به مراتب ساده‌تر مدیریت می‌شوند و در صورت نیاز به اِعمال تغییر در یکی از ماژول‌ها، اصلاً نیازی به بیلد کردن مجدد کل اپلیکیشن نخواهد بود.

### نسخه‌بندی

نسخه‌بندی کانتینرها هم یکی دیگر از مزیت‌های این فناوری است. در واقع، می‌شود برای هر کانتینر یک ورژن در نظر گرفت، تفاوت ورژن‌های مختلف با یکدیگر را مشخص ساخت و در صورت نیاز، به ورژن‌های قبلی کانتینرها اصطلاحاً Roll-back (بازگشت) کرد.

## معایب استفاده از کانتینربندی

اگرچه محبوبیت این روش از پیاده سازی بالاست و روز به روز به میزان استفاده از آن در میان برنامه نویس‌‌ها و متخصصین دوآپس بیشتر میشود؛ اما نمیتوان از عیوب موجود در این روش چشم پوشی کرد در این قسمت قصد داریم به برخی از این عیوب بپردازیم.

### امنیت

با توجه به اینکه کانتینرها مستقل از سیستم‌‌عامل نیستند و چندین کانتینر از یک سیستم‌عامل واحد استفاده می‌کند، تهدیدات امنیتی به نسبت فناوری مجازی‌سازی پررنگ‌تر است. البته برخی سرویس‌ها همچون [داکر](https://virgool.io/@arvinmoj/docker-%D8%AF%D8%A7%DA%A9%D8%B1-%DA%86%DB%8C%D8%B3%D8%AA-%D9%88-%DA%86%D8%B1%D8%A7-%D8%A7%DB%8C%D9%86%D9%82%D8%AF%D8%B1-%D9%85%D8%AD%D8%A8%D9%88%D8%A8-%D8%A7%D8%B3%D8%AA-dtz76tjt95xx)ادعا می‌کنند که کاملاً اپلکیشن‌ها را هم نسبت به یکدیگر و هم نسبت به پلتفرمی که کانتینرها روی آن اجرا می‌شوند **ایزوله** می‌کنند که این مسئله امنیت مورد نیاز را ایجاد خواهد کرد. اما باز هم این موضوع باعث میشود گروهی از برنامه نویسان تمایل خود را برای استفاده از ماشین‌های مجازی حفظ کنند.

### نیازمندی‌های سیستم

نقطه ضعف دیگری که متوجه کانتینرها است، مجدد به وابستگی آن‌ها به یک سیستم‌عامل باز می‌گردد. به عبارت دیگر، کانتینری که مبتنی بر پلتفرم لینوکس طراحی شده باشد، قابل‌اجرا روی یک پلتفرم ویندوزی نیست.

داکر برای رفع این موضوع از این امکان استفاده کرده که وقتی میخواهیم image لینوکسی بر روی ویندوز اجرا کنیم، یک ماشین مجازی لینوکس سبک اجرا میکند و کانتینر را روی آن اجرا میکند که پرفورمنس را کمتر از حالت مطلوب میکند.

### مدیریت

درست است که کانتینرها از بسیاری جهات برای محیط‌های میزبان توزیع‌شده مناسب‌تر هستند، این ویژگی نیز می‌تواند مدیریت آن‌ها را سخت‌تر کند. زمانی که صدها کانتینر در بسیاری از سرورها پخش می‌شوند، کنترل اوضاع به سرعت دشوار می‌شود. به همین دلیل است که از ابزارهای سازمان دهنده‌ای مثل Kubernetes استفاده میشود که بخش اعظم پیچیدگی ناشی از زیاد شدن کانتینرها در سیستم‌های توزیع شده را مدیریت می‌کند. با این حال، ابزار سازمان دهنده خودش یک لایه دیگر از پیچیدگی را اضافه می‌کند که شما باید راه‌اندازی، مدیریت و امنیت آن را تامین کنید.

## داکر

داکر یک پلت فرم open source که توسط Solomon Hykes به کمک مهندسان شرکت dotcloud از جمله Andrea Luzzardi و Francois-Xavier Bourlet و Jeff Lindsay(همکار مستقل) با زبان برنامه نویسی Go در سیستم عامل لینوکس نوشته شد و در 13 مارس 2014 ارائه داد.

### کاربرد

از مهمترین دغدغه‌های سرویس‌های عملیاتی تست‌، به روزرسانی‌ و نگهداری سرویس‌ها می‌باشد. به دلیل اینکه ایجاد محیطی همانند محیط عملیاتی پیاده‌سازی شده همواره پر هزینه و دشوار می‌باشد از این رو به روز‌رسانی‌ها و یا هر اقدام دیگری در آزمایشگاه‌هایی متفاوت با محیط عملیاتی انجام می‌شود که معمولا درست اعمال شده اما در محیط عملیاتی با مشکل مواجه می‌شوند. به این مشکل این را نیز اضافه کنید که ممکن است در یک پروژه چندین محیط آزمایشگاهی با تمام امکانات محیط عملیاتی برای تیم‌های متفاوت نیاز باشد. این مشکل با استفاده از داکر به صورت کامل برطرف خواهد شد. شما می‌توانید دقیقا عین همان چیزی که در محیط عملیاتی خود مورد استفاده قرار داده‌اید را در محیط‌های آزمایشگاهی و محیط‌های توسعه‌ی سرویس با کمترین زمان و هزینه و منابع دارا باشید. حتی می‌توانید کل سرویس خود را بر روی رایانه‌ی خود داشته باشید و موارد مربوط به آن را بررسی نمایید.

به عنوان مثال فرض کنید که فردی به عنوان برنامه نویس در یک شرکت برنامه را روی ماشین شخصی خود توسعه داده است و وقتی برنامه را برای مسئول devOps ارسال میکند که آن را دیپلوی کند، دچار مشکل میشود؛ زیرا نیازمندی‌‌های آن نرم افزار توسعه یافته روی کامپیوتر شخصی برنامه نویس، با سرورهای شرکت، نسخه هایی یکسان ندارند به همین دلیل فرایند دیپلوی برای این نرم افزار کاربردی سخت میشود. این درحالیست که اگه آن نرم افزار به صورت یک تصویر داکر درمی‌آمد؛ بدون نیاز به بررسی صرفا کافی بود برنامه را اجرا کنیم تا خود به خود نیازمندی هایش را پیدا کند و اجرا شود.

### مقیاس پذیری داکر

مقیاس‌پذیری از مهمترین قابلیت‌های داکر و میکروسرویس‌ها می‌باشد. تنها این دلیل به اندازه کافی ایجاب می‌کند تا برای پروژه‌ی خود از سرویس داکر استفاده کنید. با استفاده از داکر به خوبی می‌توان مقیاس‌پذیری را در تمام سطح پروژه‌ی خود پیاده‌سازی کرده و مهمتر از آن اینکه می‌توان به خوبی مقیاس‌پذیری خودکار (Auto Scaling) را ایجاد کرد. با توجه به این موضوع در هنگام پیک درخواست‌ و نیاز اساسی به منابع میزان آنها افزایش یافته و پس از برطرف شدن نیاز آن منابع کاهش یابد و به صورت خودکار جمع‌آوری شود. که این موضوع هم در کیفیت سرویس‌دهی شما بسیار موثر خواهد بود و هم هزینه‌های شما را به شدت کاهش خواهد داد.

### قابل حمل بودن داکر

یکی دیگر از قابلیت‌های مهم داکر قابلیت قسمت‌بندی کردن پروژه می‌باشد که می‌توان با استفاده از آن پروژه‌‌ای بزرگ را به بخش‌های کوچک‌تر تقسیم کرد که هر بخش را به صورت مستقل پیاده‌سازی و اجرا کرد. سپس تمام بخش‌های را کنار هم قرار داده و سرویس اصلی را ایجاد نمود. این موضوع در پروژه‌های بزرگ به شدت موثر بوده و سرعت اجرا و پیاده‌سازی آن را افزایش می‌دهد. از دیگر قابلیت‌های مهم قابل حمل بودن می‌باشد. بر این اساس شما می‌توانید کل پروژه را در محیط دیگری آماده نمایید و سپس آن را به محیط عملیاتی خود منتقل کنید. این قابلیت امکان به اشتراک‌گذاری سرویس‌های پیاده‌سازی شده را نیز فراهم می‌آورد.

### حجم کم تصاویر در داکر

سرویس داکر با استفاده از قابلیت‌های Namespace و CGroups کرنل لینوکس، امکانی را فراهم می‌آورد که با توجه به مشترک بودن کرنل لینوکس ماشین میزبان بین تمام کانتینرها، محیط‌های ایزوله و مستقلی را ایجاد کند تا پروسه‌ها بدون تاثیر بر روی یکدیگر هر کدام به صورت مجزا در کانتینر خود اجرا شود. این امکان به شدت در حجم محصول نهایی موثر بوده و آن را به مراتب کم حجم‌تر و سبک‌تر خواهد کرد. نکته‌ی مهم دیگه اینکه با توجه به قابلیت لایه‌‌ای بودن ایمیج‌ها از لایه‌های تکراری به صورت اشتراکی استفاده می‌شود و این موضوع نیز به شدت در حجم کانتینر‌ها موثر خواهد بود. سبک‌تر شدن کانتینرها باعث می‌شود تا سرعت بیشتری را نیز شاهد باشیم.

## خلاصه

در پاسخ به این پرسش که «آیا کانتینرها کاملاً جای ماشین‌های مجازی را خواهند گرفت؟» باید اعتراف کرد که جواب شفافی وجود ندارد اما از شواهد امر برمی‌آید که چنین چیزی حداقل به یک دلیل در آینده نزدیک امکان‌پذیر نخواهد بود و آن هم اینکه چیزی که کماکان فناوری مجازی‌سازی را محبوب نگاه داشته این است که به دلیل ایزوله‌تر بودن ماشین‌های مجازی نسبت به کانتینرها، برخی بر این باورند که ماشین‌های مجازی **امن‌تر** می‌باشند اما به هر حال مزیت‌های کانتینرها به اندازه‌ای بوده که امروزه کسب‌وکارهای بزرگی همچون مایکروسافت، گوگل، آمازون و غیره شروع به سرمایه‌گذاری روی این فناوری کرده‌اند.

# فصل چهارم مقایسه و بررسی ماشین مجازی و سیستم‌‌های مبتنی بر کانتینرمقایسه و بررسی ماشین مجازی و سیستم‌‌های مبتنی بر کانتینر

هم ماشین مجازی و هم کانتینربندی روش هایی هستند که کمک بسیار زیادی در کنترل و مدیریت نرم افزار و سخت افزار به شما میدهند. کانتینر فناوری جدید و به روزی است و ماشین‌‌های مجازی محبوب در مراکز داده با اندازه‌‌های بزرگ هستند.

اگر به دنبال بهترین راه حل برای مدیریت سرویس هایتان بر روی فضای ابری هستید بایستی شناخت مناسبی از فن‌آوری‌‌های مجازی سازی، یعنی ماشین مجازی و کانتینر داشته باشید. در این فصل قصد داریم که این دو فناوری را در موارد مختلف با هم مقایسه کنیم.

## انزوا

ماشین مجازی امکان جداسازی کامل از سیستم‌عامل میزبان و سایر ماشین‌‌های مجازی را فراهم می‌کند. این زمانی مفید است که یک مرز امنیتی قوی حیاتی لازم باشد؛ مثلا میزبانی برنامه‌‌های کاربردی شرکت‌‌های رقیب در همان خوشه از سرور‌‌ها را داشته باشیم.

به طور معمول انزوای ایجاد شده در حالت استفاده از کانتنربندی ضعیف تر است و به علت جدا نبودن هسته سیستم عامل میزان انزوا به نسبت ماشین مجازی کمتر است، هر چند که با استفاده از فناوری های جدید مثل داکر تاحد خوبی امنیت لازم تامین میشود)

بنابراین میتوان گفت که امنیت در ماشین مجازی بیشتر ایجاد میشود.

## سیستم عامل

که ماشین های مجازی سیستم عامل کاملی را درون خود دارند و از جمله هسته آن را اجرا میکنند. به همین دلیل میزان نیاز به منابع ازقبیل پردازنده، حافظه اصلی و غیره، در این فناوری بیشتر است.

هر کانتینر در user mode اجرا میشود، بنابراین می‌تواند با استفاده از منابع سیستم کم‌تر، تنها خدمات مورد نیاز برای برنامه شما را از طریق هسته سیستم عامل اجرا کند.

## عملکرد

از آن جایی که ماشین های مجازی از هسته اصلی استفاده نمیکنند و خودشان سیستم عاملی برای خودشان دارند، میزان اختصاص منابع به ماشین مجازی محدود است؛ در نتیجه عملکرد ماشین مجازی نیست محدود میشود.

کانتینر ها با توجه به این که از هسته سیستم عامل میزبان استفاده میکنند؛ محدودیتی در عملکرد و منابع ندارند.

## حجم تصاویر

همانطور که قبلا نیز گفته شد با توجه به این که ماشین های مجازی یک سیستم عامل را به طور کامل دارند بنابرین میزان حجم تصاویر آن ها به طور قابل توجهی از کانتینر ها بیشتر است.

## زمان راه اندازی

به علت محدودیت های موجود در ماشین مجازی زمان راه‌اندازی یا همان startup time در این فناوری به نسبت فناوری کانتینربندی بیشتر است. یعنی مثلا اگر ماشین مجازی در چند دقیقه راه اندازی شود، این موضوع برای کانتینر ها در چند ثانیه خواهد بود.

|  |  |
| --- | --- |
| کانتینر | ماشین مجازی |
| حجم تصاویر کم | حجم تصاویر زیاد |
| عملکرد در سطح ماشین میزبان | عملکرد محدود |
| اجرا بر سیستم عامل میزبان | اجرا بر سیستم عامل محلی |
| سطح انتزاع سیستم عامل | سطح انتزاع سخت افزاری |
| زمان راه اندازی در سطح ثانیه | زمان راه اندازی در سطح دقیقه |
| حافظه اصلی کمتری نیاز دارد | حافظه اصلی بیشتری نیاز دارد |
| امنیت کمتر | امنیت بیشتر |

جدول 4-1 خلاصه مقایسه ماشین مجازی و کانتینر

# فصل پنجم جمع‌بندی و نتیجه‌گیری و پیشنهاداتجمع‌بندی و نتیجه‌گیری

امرازه تحقیقات زیادی در زمینه‌ی خلاصه‌سازی متن در جریان است. ما نیز سعی کردیم در این مقاله یکی از این الگوریتم‌های موجود را بهینه‌تر کنیم. که نتایج آن در فصل قبل قابل مشاهده بود. ما با استفاده از چهار ویژگی عنوان شده برای متن و به کمک ماشین بولتزمن محدود نتایج الگوریتم خلاصه‌سازی را با کمک یادگیری عمیق بهینه کردیم. به این صورت که ابتدا قواعد ماشین بولتزمن محدود را توسط الگوریتم یادگیری عمیق براز کردیم. سپس داده‌ها و ویژگی‌ها را در سطوح مختلف ماشین بولتزمن محدود پردازش کرده و خلاصه‌ی متن را بر این اساس تولید کردیم. در نهایت با آزمایش این الگوریتم بر رای دو مجموعه سند، کارآیی آن را سنجیدیم و دیدیم که بهتر از الگوریتم موجود برای خلاصه‌سازی متن عمل کرد. بهبود آینده‌نگرانه‌ی رایکرد پیشنهادی با در نظر گرفتن ویژگی‌های مختلف و افزودن چند لایه‌ی مخفی به الگوریتم ماشین بولتزمن محدود، حاصل می‌شود.

**پیشنهادات**

در انتها پیشنهاد می‌دهیم در زمینه‌ی افزایش کارآیی راش خلاصه‌سازی متن پیشنهادی مطالعه کنید. این بهبود می‌تواند با کاهش بیشتر مقدار فراخوانی‌ها و یا افزایش دقت الگوریتم به دست آید. همچنین می‌توان علاوه بر چهار ویژگی نام برده در مقاله برای هر جمله، ویژگی‌های بیشتری برای هر جمله بیابید تا عمل خلاصه‌سازی با دقت و کیفیت بیشتر و بالاتری صورت پذیرد.

# منابع و مراجع

Darling, William M, and Fei Song. "Probabilistic Document Modeling for Syntax Removal in Text Summarization." Paper presented at the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, 2011.

Das, Dipanjan, André FT %J Literature Survey for the Language Martins, and Statistics II course at CMU. "A Survey on Automatic Text Summarization." 4 (2007): 192-95.

Duraiswamy, K. "An Approach for Text Summarization Using Deep Learning Algorithm." (2014).

Goldstein, Jade, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. "Multi-Document Summarization by Sentence Extraction." Paper presented at the Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization, 2000.

Gong, Yihong, and Xin Liu. "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis." Paper presented at the Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001.

Hovy, Eduard. "Text Summarization." In *The Oxford Handbook of Computational Linguistics 2nd Edition*, 2003.

Kogilavani, A, and P %J American Journal of Applied Sciences Balasubramanie. "Sentence Annotation Based Enhanced Semantic Summary Generation from Multiple Documents." 9, no. 7 (2012): 1063.

Lin, Chin-Yew, and Eduard Hovy. "The Automated Acquisition of Topic Signatures for Text Summarization." Paper presented at the Proceedings of the 18th conference on Computational linguistics-Volume 1, 2000.

Mani, Inderjeet. *Advances in Automatic Text Summarization.* MIT press, 1999.

———. "Recent Developments in Text Summarization." Paper presented at the Proceedings of the tenth international conference on Information and knowledge management, 2001.

Neto, Joel Larocca, Alex A Freitas, and Celso AA Kaestner. "Automatic Text Summarization Using a Machine Learning Approach." Paper presented at the Brazilian Symposium on Artificial Intelligence, 2002.

Ou, Shiyan, Christopher Soo-Guan Khoo, and Dion H %J Journal of information science Goh. "Design and Development of a Concept-Based Multi-Document Summarization System for Research Abstracts." 34, no. 3 (2008): 308-26.

Patil, Kaustubh, and Pavel %J Int. J. Comput. Sci. Inform. Syst Brazdil. "Text Summarization: Using Centrality in the Pathfinder Network." 2, no. 18-32 (2007).

Tieleman, Tijmen. "Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient." Paper presented at the Proceedings of the 25th international conference on Machine learning, 2008.

Yousefi-Azar, Mahmood, and Len %J Expert Systems with Applications Hamey. "Text Summarization Using Unsupervised Deep Learning." 68 (2017): 93-105.

Zhang, Yi, Dingding Wang, and Tao Li. "Idvs: An Interactive Multi-Document Visual Summarization System." Paper presented at the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2011.