کوبرنتیس

Kubernetes

نام ارائه دهنده	رحمت اله انصاری	عرفان قربانی
شماره دانشجویی	۹۹۱۲۳۷۷۳۳۱	۹۹۲۲۳۷۷۳۰۹
ایمیل	Rahmat2022a@gmail.com	-

فهرست مطالب

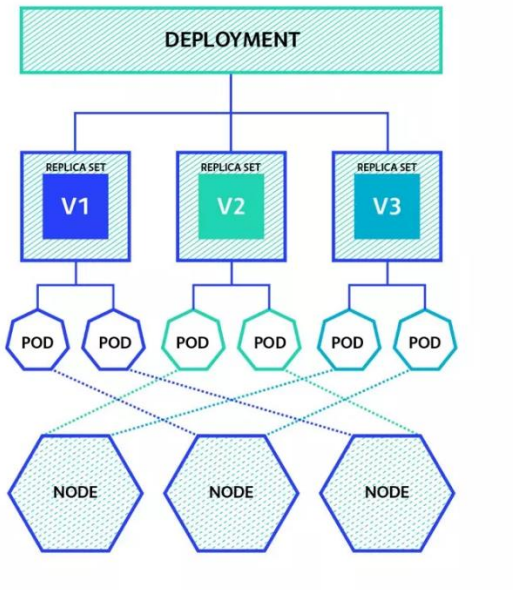
کوبرنتیس چیست و چه کاربردهایی دارد؟

کانتینرها با مزایای متعددی مثل اجرای پروژه فارغ از زیرساخت محبوبیت زیادی پیدا کرده‌اند. پلتفرم کوبرنتیس را گوگل معرفی کرده و به یکی از ابزارهای اصلی برای استقرار و مدیریت برنامه‌های کانتینری تبدیل شده است.

کوبرنتیس چیست؟

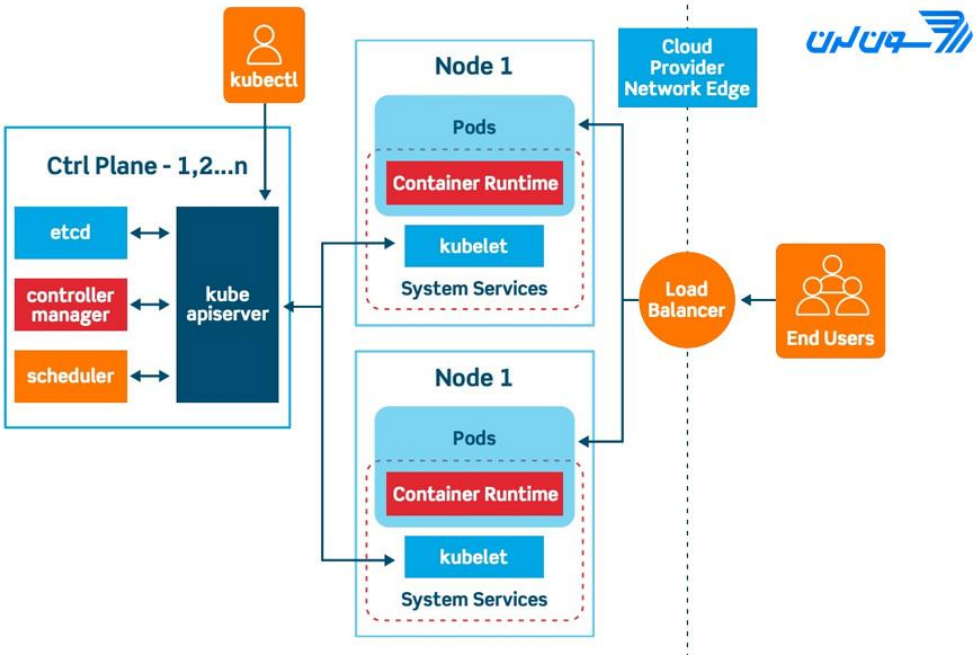
کوبرنتیس (Kubernetes) پلتفرمی متن‌باز و توسعه‌پذیر و پرتابل برای پیاده‌سازی و مدیریت برنامه‌های کانتینری است. کوبرنتیز با زیست‌بوم گسترده‌اش، فرایندهای خودکارسازی و پیکربندی اپلیکیشن‌ها را ساده می‌کند و زمان و منابع لازم برای اجرای عملیات‌ها را کاهش می‌دهد. نکته دیگر اینکه گستردگی خدمات و پشتیبانی و ابزارهای آن باعث شده است تا بین توسعه‌دهندگان به‌سرعت محبوبیت پیدا کند.

اگر بخواهیم به‌زبان ساده کوبرنتیز را توضیح دهیم باید بگوییم کوبرنتیز اجرا و مدیریت کانتینرهای مختلف را در سرورهای متفاوت که در یک پایگاه داده یا چندین پایگاه قرار گرفته‌اند را بر عهده می‌گیرد. در کوبرنتیز کانتینرهای مختلفی که مشترکاً برنامه کاربردی خاصی را شامل می‌شوند در حالت جداگانه و مستقل تحت عنوان پاد (Pod) دسته‌بندی خواهند شد. این کار فرآیند مدیریت و شناسایی آن‌ها را ساده‌تر می‌کند.



نام Kubernetes ریشه یونانی دارد و به‌معنی «سکان‌دار» یا «خلبان» است. حرفه‌ای‌ها این ابزار را K8S هم می‌نامند که به حروف ابتدایی و انتهایی کوبرنتیس و فاصله هشت‌حرفی بین آن‌ها اشاره می‌کند. کوبرنتیس در بخش زیرساخت ابری گوگل توسعه یافته و در سال 2014 به‌صورت متن‌باز ارائه شده است. این ابزار نتیجه تجربه ۱۵ساله گوگل در مدیریت اپ‌های کانتینری است و جامعه متن‌باز هم در توسعه آن مشارکت کرده‌اند.

گوگل یکی از نخستین مشارکت‌کنندگان در فناوری کانتینر لینوکس بود. این شرکت برای استقرار کانتینرها که تعدادشان به دومیلیارد در هفته می‌رسد، از پلتفرم Borg کمک می‌گیرد که پدر Kubernetes محسوب می‌شود.



تصویر بالا یک نما از چگونگی کارکرد کوبرنتیز را نشان می‌دهد. اگر با مفاهیم آن ناآشنا هستید نگران نباشید، جلوتر هر جز را به صورت مفصل توضیح خواهیم داد.

کوبرنتیس چگونه کار می‌کند؟

برای درک بهتر مزایای کوبرنتیز، باید نگاهی به تاریخچه استقرار نرم‌افزارها بیندازیم:

- عصر استقرار سنتی: در این دوران، نرم‌افزار روی سرورهای فیزیکی اجرا می‌شد و چون راهی برای تعیین محدودیت‌های سخت‌افزاری وجود نداشت، مشکل اختصاص منابع پیش می‌آمد. برای مثال، حین اجرای هم‌زمان چند اپ روی سرور، یکی از برنامه‌ها تمام منابع را ازآنِ خود می‌کرد و اجرای دیگر اپ‌ها به مشکل می‌خورد. یکی از راه‌های حل این مشکل استفاده از چند سرور است؛ اما هزینه زیادی را به شرکت‌ها تحمیل می‌کند.
- عصر استقرار مجازی: در این دوران، با اجرای ماشین‌های مجازی روی یک سرور منابع مشخصی به هر نرم‌افزار تعلق می‌گرفت. این روش نه‌تنها امنیت را افزایش داد؛ بلکه امکان اضافه‌کردن و آپدیت آسان اپ‌ها مقیاس‌پذیری را نیز بهبود بخشید. هر ماشین مجازی اجزا و سیستم‌عامل و منابع خاص خودش را دارد.
- عصر استقرار کانتینر: کانتینرها شبیه ماشین مجازی هستند؛ اما سیستم‌عامل بین آن‌ها مشترک است؛ به‌همین‌دلیل، بار کمتری برای سخت‌افزار دارند. کانتینر نیز همچون ماشین مجازی فایل‌های سیستم خودش را دارد و بخشی از پردازنده و حافظه و فضای ذخیره‌سازی را دراختیار می‌گیرد.

کوبرنتیس برای اجرای کارها به کلاستر اتکای زیادی می‌کند. کلاستر Kubernetes را می‌توانید در دو بخش تصور کنید:

۱. سطح کنترل؛
۲. ماشین‌های محاسباتی یا گره‌ها.

هر گره یا نود محیط خاص خودش را دارد که می‌تواند ماشینی مجازی یا فیزیکی باشد. این گره پادهایی را اجرا می‌کند که از کانتینرها تشکیل شده است. سطح کنترل نیز مسئولیت نگه‌داری حالت دلخواه کلاستر را برعهده دارد؛ مثلاً اینکه کدام اپلیکیشن‌ها اجرا شوند و این اپ‌ها از چه ایمیج کانتینری استفاده کنند. کوبرنتیز مثل هر اپ دیگری در سیستم‌عامل اجرا می‌شود و با پادهای کانتینر اجراشده در گره‌ها تعامل دارد.

سطح کنترل Kubernetes دستورها را از مدیر سیستم یا تیم DevOps دریافت و به ماشین‌های محاسباتی ارسال می‌کند. در این مسیر، محاسبات زیادی انجام می‌شود تا بهترین گره برای تسک مدنظر مشخص شود. پس از تعیین گره، برای اجرای وظایف مدنظر منابع به آن اختصاص داده می‌شود. حالت دلخواه کلاستر Kubernetes

مواردی مانند این‌ها را مشخص می‌کند: کدام اپلیکیشن‌ها یا وظیفه‌ها باید اجرا شوند, این اپ‌ها از چه ایمیجی باید استفاده کنند, چه منابعی باید به آن‌ها اختصاص یابد و دیگر جزئیات پیکربندی.

این موارد از دیدگاه زیرساختی تغییر کوچکی در چگونگی مدیریت کانتینرهاست. شما بدون درگیرشدن در جزئیات هر کانتینر یا گره مجزا, تنها در سطوح بالا کانتینرها را کنترل می‌کنید. کار اصلی شما پیکربندی کوبرنتیس و تعریف گره‌ها و پادها و کانتینرهای درون آن‌هاست. هماهنگ‌سازی این بخش‌ها هم برعهده Kubernetes است. کوبرنتیس ازنظر پلتفرم اجرا دستتان را بازمی‌گذارد. از سرور فیزیکی گرفته تا ماشین مجازی و سرویس‌های ابری عمومی یا خصوصی و محیط ابری هیبرید همه قابلیت اجرای کوبرنتیس را دارند.

کانتینر چیست؟

کانتینر را مانند بسته‌ای نرم‌افزاری در نظر بگیرید که تمام فایل‌های سیستمی و کتابخانه و کدهای یک پروژه را در خود دارد. این کانتینر به اعضای تیم اجازه می‌دهد تا بدون نگرانی درباره سیستم‌عامل یا زیرساخت, پروژه را در محیط‌های مختلف اجرا و تست کنند.

کانتینرها مستقل از زیرساخت عمل می‌کنند و می‌توان آن‌ها را در توزیع‌های مختلف سیستم‌عامل یا زیرساخت ابری جابه‌جا کرد. ازجمله مزایای کانتینر درمقایسه‌با ماشین مجازی می‌توان به این‌ها اشاره کرد:

- وابستگی نداشتن به زیرساخت در مراحل توسعه و تست: کانتینرها فارغ از محیط زیرساخت در لپ‌تاپ, سرور, محیط ابری و حتی هیبرید به یک شکل اجرا می‌شوند.
- توزیع‌های پرتابل در OS و ابر: در ویندوز, اوبونتو, CoreOS,RHEL, اکثر سرویس‌های ابری و... اجرا می‌شوند.
- کاهش بار روی زیرساخت: استفاده از سیستم‌عامل مشترک و سبک‌بودن بسته‌های کانتینر درمقایسه‌با ماشین مجازی.
- تولید و استقرار سریع اپلیکیشن: سهولت و بهره‌وری چشمگیر تولید فایل ایمیج کانتینر درمقایسه‌با ایمیج ماشین مجازی.

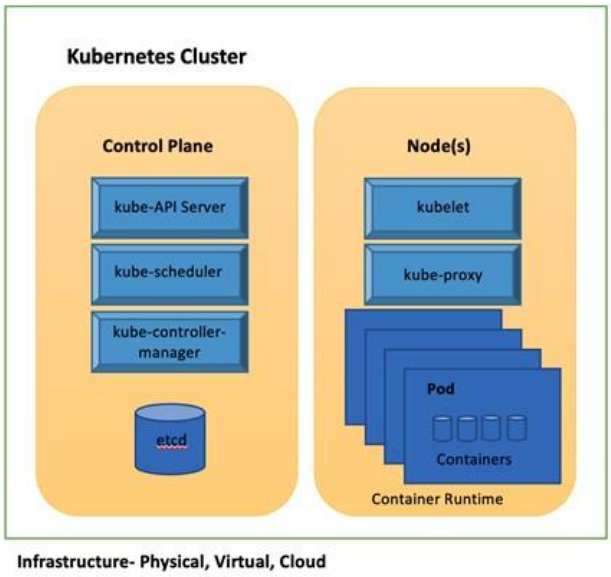
- توسعه و یکپارچه‌سازی و استقرار پیوسته نرم‌افزارها: تولید و استقرار ایمیج‌های کانتینر متعدد و مطمئن با امکان بهینه‌سازی سریع و کارآمد.
- جداسازی دغدغه‌های بخش توسعه و عملیات: جداسازی اپ‌ها از زیرساخت به‌واسطه تولید ایمیج‌های کانتینر در زمان تولید یا انتشار به‌جای زمان استقرار.
- امنیت فراوان: کانتینر نرم‌افزارها را از یکدیگر ایزوله و دسترسی به داده‌ها را محدود می‌کند.

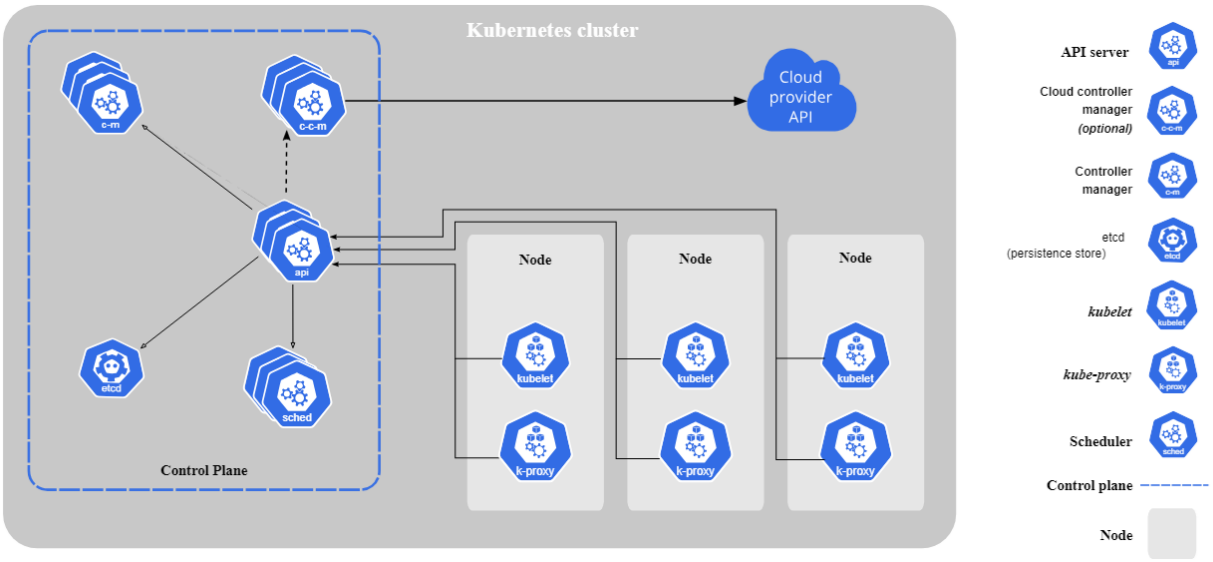
کلاستر کوبرنتیس چیست؟

زمانی‌که کوبرنتیس را مستقر می‌کنید، یک کلاستر تشکیل می‌شود. تمامی اجزا و قابلیت‌ها و بار کاری Kubernetes در کلاستر پیکربندی می‌شود. کوبرنتیز امکان مدیریت آسان و کارآمد این کلاسترها را برای شما فراهم می‌کند.

هر کلاستر از دو بخش اصلی ماشین محاسباتی (گره کارگر) و بخش کنترلی (گره اصلی) تشکیل می‌شود و باید حداقل یک گره اصلی و یک گره کارگر داشته باشد. کلاستر امکان اجرای کانتینرها در انواع محیطهای میزبان مثل ماشین مجازی، فیزیکی، ابری و هیبریدی را فراهم می‌کند؛ ازاین‌رو، کوبرنتیز پلتفرمی ایده آل برای میزبانی اپ‌های ابری نیتیو مثل استریم آنلاین داده است که به مقیاس‌پذری سریع نیاز دارند.

از کوبرنتیس می‌توانید برای پیاده‌سازی و مدیریت نرم‌افزارهای پیچیده تحت عنوان کانتینر استفاده کنید. این ابزار متن باز به شما کمک می‌کند تا فرایند پیکربندی برنامه‌ها سریع‌تر و ساده‌تر شود. از ویژگی‌های مهم کوبرنتیس می‌توان به بهینه‌سازی فرایند توسعه اپلیکیشن‌ها اشاره کرد.



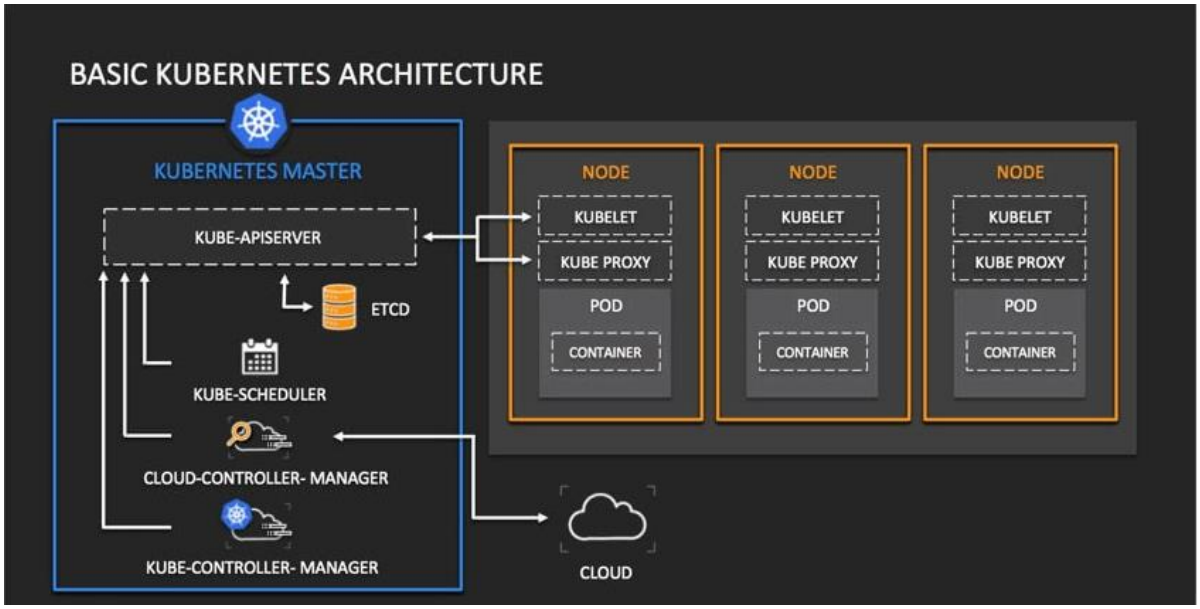


مربوط به سایت اصلی کوبرنتیز (+)

آشنایی با معماری کوبرنتیز

حال که با مفاهیم اصلی آشنا شدید، در این بخش ساختار و معماری اصلی کوبرنتیز را بررسی می‌کنیم. خوشه کوبرنتیز شامل گره‌های رایانه‌ای زیادی است که به گره‌های اصلی و کارگر تقسیم می‌شوند. با کوبرنتیز،

می‌توانید برنامه‌های نرم‌افزاری خود را بر روی هزاران گره اجرا کنید، به‌گونه‌ای که انگار یک رایانه واحد و عظیم هستند.

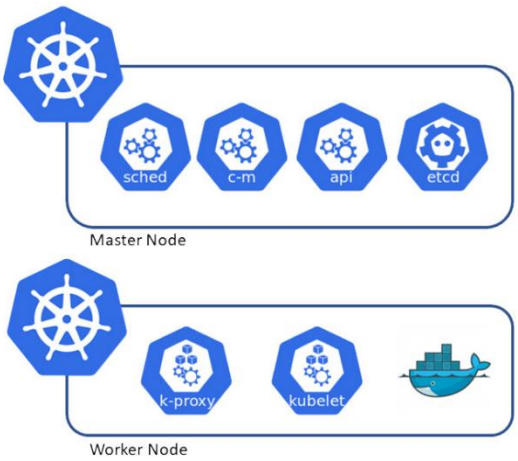


گره اصلی

هر خوشه شامل حداقل یک گره اصلی است که میزبان بخش کنترل کوبرنتیز است که کل سیستم کوبرنتیز را کنترل و مدیریت می‌کند. گره اصلی تمام فرآیندهای حیاتی را که در برنامه کوبرنتیز شما اجرا می‌شوند نگه می‌دارد. چند گره کارگر به این گره اصلی متصل هستند، اما ارزش کمتری نسبت به گره اصلی برای خوشه دارند. نکته مهم این است که اگر دسترسی به گره اصلی را از دست بدهید، به طور کلی دسترسی به خوشه را از دست خواهید داد.

گره کارگر

گره‌های کارگر همان چیزی هستند که برنامه‌های کاربردی را اجرا می‌کنند. هر گره کارگر شامل چندین کانتینر از برنامه‌های مختلف است که روی آن اجرا می‌شوند. آنها بزرگ هستند و حاوی منابع زیادی هستند، و در نتیجه، گره کارگر بیشتر حجم کاری برنامه کوبرنتیز شما را انجام می‌دهد. گره‌های کارگر دارای یک فرآیند Kubelet یا همان خط فرمان هستند که برای اجرا بر روی آنها طراحی شده است.



کاربردهای کوبرنتیس چیست؟

کاربرد اصلی کوبرنتیس پیاده‌سازی و مدیریت آسان برنامه‌های کانتینر شده است. اگر به‌دنبال بهینه‌سازی فرایند توسعه اپ برای ابر باشید، این ابزار پلتفرم قدرتمندی را برای زمان‌بندی و اجرای کانتینرها در کلاسترهای ماشین مجازی یا فیزیکی دراختیارتان می‌گذارد.

Kubernetes در پیاده‌سازی کامل زیرساخت مبتنی‌بر کانتینر در محیط توسعه به شما کمک می‌کند. ازآن‌جاکه این پلتفرم برای خودکارسازی تسک‌های عملیاتی طراحی شده است، به شما اجازه می‌دهد تا بسیاری از کارهای دیگر را نیز روی کانتینرها انجام دهید. توسعه‌دهندگان با ابزارهای خاص الگوهای کوبرنتیس می‌توانند برنامه‌های ویژه زیرساخت ابری را به‌عنوان پلتفرم ران‌تایم تولید کنند. دیگر کاربردهای کوبرنتیز عبارت‌اند از:

- هماهنگ‌سازی کانتینرها در چندین سیستم میزبان
- کنترل و خودکارسازی روند استقرار و به‌روزرسانی اپلیکیشن‌ها
- ارتقای حافظه برای اجرای اپ‌های حالتمند (Stateful)
- افزایش آنی مقیاس اپ‌های کانتینری و منابع آن‌ها
- اطمینان از اجرای صحیح و دقیق اپ‌های مستقر
- بررسی و اصلاح خودکار اپ‌ها با قابلیت‌های ارتقا و مقیاس‌پذیری خودکار

کوبرنتیس برای ارائه کامل این خدمات به پروژه‌های متن‌باز دیگر اتکا دارد. تعدادی از این اپ‌ها بدین‌شرح‌اند:

- رجیستری ازطریق پروژه‌هایی مثل Docker Registry
- شبکه‌سازی ازطریق پروژه‌های OpenvSwitch و مسیریابی لبه هوشمند
- تله‌متری ازطریق Kibana و Hawkular و Elastic
- امنیت ازطریق پروژه‌هایی مثل LDAP, SELinux,RBAC و OAUTH با لایه‌های چندمستأجری (Multitenancy)
- خودکارسازی با افزودن پلی‌بوک‌های Ansible برای نصب و مدیریت چرخه عمر کلاستر

چرا به کوبرنتیس نیاز داریم؟

کانتینرها روشی ایدئال برای باندل‌کردن و اجرای اپلیکیشن‌ها به‌شمار می‌آیند. در محیط توسعه, کانتینرها را باید به‌دقت مدیریت کنید و مطمئن شوید از دسترس خارج نشده‌اند. برای مثال, اگر کانتینری خاموش شود, کانتینر دیگری باید شروع شود. آیا سپردن این فرایند نظارتی دقیق و مکرر به یک سیستم بهتر نیست؟

این‌جاست که Kubernetes برای نجات شما وارد صحنه می‌شود. این ابزار محبوب فریم‌ورکی را برای اجرای انعطاف‌پذیر سیستم‌های توزیع‌شده دراختیارتان می‌گذارد. کوبرنتیس روی مقیاس‌پذیری و اجرای صحیح اپلیکیشن شما نظارت و الگویی را برای استقرار و توسعه فراهم می‌کند. از دیگر کاربردهای کوبرنتیس می‌توان به این‌ها اشاره کرد:

- سرویس جست‌وجو و متعادل‌سازی بار: کوبرنتیس می‌تواند کانتینر را ازطریق نام DNS یا آدرس IP پیدا کند. اگر ترافیک ارسالی به یک کانتینر زیاد باشد, ترافیک به بخش‌های دیگر توزیع و با متعادل‌سازی بار از ناپایداری اپلیکیشن جلوگیری می‌شود.
- هماهنگ‌سازی حافظه: کوبرنتیس قابلیت نصب خودکار حافظه براساس اولویت‌های شما را دارد. این حافظه می‌تواند محلی, ابری و... باشد.
- تعیین حالت دلخواه اپلیکیشن: حالت دلخواه خود را برای کانتینرهای مستقر توصیف کنید تا Kubernetes با نرخ مشخصی وضعیت موجود را به حالت دلخواهتان تغییر دهد. برای مثال, می‌توانید

Kubernetes را برای ایجاد کانتینرهای جدید و حذف کانتینرهای موجود و اختصاص منابع به موارد جدید تنظیم کنید.

- استفاده بهینه از سخت‌افزار: برای افزایش حداکثری منابع دردسترس برنامه‌ها باید مشخص کنید که هر کانتینر به چه میزان توان پردازش و RAM نیاز دارد.
- خودترمیمی: کوبرنتیس کانتینرهای دچار خطا را ری‌استارت و جایگزین می‌کند و مواردی که براساس وضعیت دلخواه کاربر نیستند, خاتمه می‌دهد و آن‌ها را تا زمانی‌که کاربردی نباشند, به کلاینت نمی‌فرستد. بدین‌ترتیب, فرایندی خودترمیمی ایجاد می‌شود.
- مدیریت داده‌های حساس و پیکربندی اپ‌ها: Kubernetes امکان ذخیره‌سازی داده‌های حساس, ازجمله رمزهای عبور و توکن‌های OAuth و کلیدهای SSH را برایتان فراهم می‌کند. بدون نیاز به بازسازی فایل‌های ایمیج کانتینر و افشای این داده‌ها در پیکربندی استک, می‌توانید این اطلاعات را مستقر و به‌روزسانی کنید.
- CI/CDساده شده Kubernetes :به‌طور یکپارچه با شیوه‌های یکپارچه‌سازی/استقرار مستمر (CI/CD) ادغام می‌شود که فرآیند ساخت, آزمایش و استقرار برنامه‌ها در محیطهای تولید را خودکار می‌کند. با گنجاندن Kubernetes در خطوط لوله CI/CD , سازمان‌ها می‌توانند گردش‌های کاری مقیاس‌پذیر و قابل انطباق ایجاد کنند که به صورت پویا با تغییرات بار کاری تنظیم می‌شوند و فرآیند کلی تحویل برنامه را ساده می‌کنند.

برای اطلاعات بیشتر!

پروتکل OAuth یک استاندارد فنی برای مجوز دادن به کاربران است. در واقع OAuth پروتکلی برای انتقال مجوز از یک سرویس به سرویسی دیگر بدون به اشتراک گذاشتن اعتبار واقعی کاربر مانند نام کاربری و رمز عبور است. با استفاده از پروتکل OAuth یک کاربر می تواند وارد یک پلتفرم شده و سپس مجاز به انجام اقدامات و مشاهده داده ها در پلتفرمی دیگر باشد.

OAuth انتقال مجوز از برنامه ای به برنامه دیگر را صرف نظر از اینکه چه برنامه ای هستند ممکن می کند. OAuth یکی از رایج ترین روش هایی است که برای انتقال مجوز از یک سرویس احراز هویت یکپارچه (SSO) به یک سامانه دیگر استفاده می شود. پروتکل های دیگر نیز می توانند این کار را انجام دهند اگرچه پروتکل OAuth یک از پرکاربردترین آنها است.

تصور کنید هنگامی که صاحب خانه در خانه نیست بازدید کننده ای به خانه می آید و مالک به جای ارسال کلید برای بازدید کننده کدی موقتی می فرستد تا وارد صندوقی شود که کلید داخل آن است. OAuth نیز به روشی مشابه کار می کند. در OAuth یک سامانه به سامانه دیگر , به جای ارسال اعتبارنامه کاربر برای اجازه دسترسی به وی، یک توکن دسترسی (authorization token) ارسال می کند.

دیدگاه‌های نادرست درباره کوبرنتیس

کاربردهای گسترده کوبرنتیز باعث ایجاد برخی دیدگاه‌های نادرست درباره آن شده است. پیش از هرچیز باید بدانید که کوبرنتیس سیستم PaaS (پلتفرم به‌عنوان سرویس) سنتی و همه‌شمول نیست؛ چون این ابزار به‌جای سخت‌افزار در سطح کانتینر عمل و برخی قابلیت‌های شبیه به PaaS مثل استقرار و مقیاس‌بندی و متعادل‌سازی بار را ارائه می‌کند.

باوجوداین, Kubernetes کاملاً یکپارچه نیست و قابلیت‌هایش انتخابی و گزینشی هستند. درواقع, این ابزار بلاک‌های زیربنایی را برای تولید پلتفرم توسعه‌دهنده فراهم می‌کند؛ اما برای کاربر حق انتخاب قائل می‌شود و انعطاف‌پذیر است.

ارکستراسیون کانتینری چیست؟

مفهوم مهم دیگری که در مورد اینکه کوبرنتیز چیست حائز اهمیت است ارکستراسیون کانتینری است. ارکستراسیون کانتینر به اتوماسیون وظایف مختلف مربوط به مدیریت و اجرای بارهای کاری و خدمات کانتینری اشاره دارد. ارکستراسیون کانتینری عملیات لازم برای چرخه حیات کانتینرها, از جمله تهیه, استقرار, مقیاس بندی, شبکه‌سازی و متعادل‌سازی بار را ساده می‌کند.

با ارکستراسیون کانتینر, این وظایف توسط یک پلتفرم یا ابزار اختصاصی, مانند Kubernetes, خودکار و مدیریت می‌شوند. این پیچیدگی های مدیریت کانتینرهای فردی را انتزاعی کرده و امکان مدیریت کارآمد استقرار کانتینر در مقیاس بزرگ را فراهم می‌کند. پلتفرم‌های هماهنگ‌سازی کانتینر وظایفی مانند زمان‌بندی کانتینرها در منابع موجود, افزایش یا کاهش بر اساس تقاضا, مدیریت شبکه و ارتباط بین کانتینرها, توزیع ترافیک در کانتینرها و مدیریت عملیات چرخه حیات کانتینر مانند بررسی سلامت و به‌روزرسانی‌های چرخشی را انجام می‌دهند.

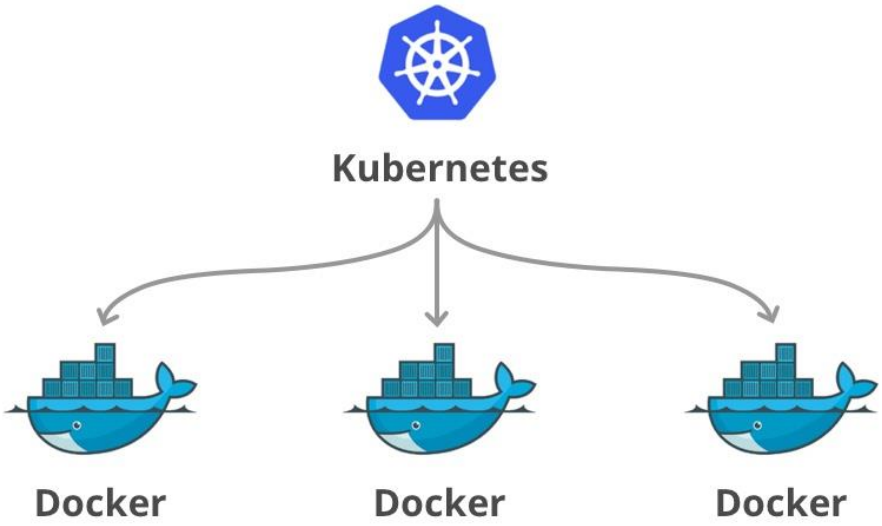
با خودکار کردن این وظایف, هماهنگ‌سازی کانتینر, مدیریت و عملکرد برنامه‌های کاربردی کانتینری را ساده‌تر کرده و استقرار, مقیاس بندی و مدیریت مؤثر آن‌ها را آسان‌تر می‌کند. این ارکستراسیون کانتینری استفاده از

منابع را بهبود می‌بخشد، در دسترس بودن بالا و تحمل خطا را امکان‌پذیر کرده و زیرساختی انعطاف‌پذیر و مقیاس‌پذیر برای اجرای بارهای کاری کانتینری فراهم می‌کند.

این مزایا، Kubernetes را به یک پلتفرم ترجیحی برای استقرار، مدیریت و مقیاس بندی برنامه‌های کانتینری، توانمندسازی سازمان‌ها برای دستیابی به قابلیت حمل، کارایی هزینه، مقیاس‌پذیری و ادغام یکپارچه با سیستم‌های موجود تبدیل می‌کند.

مقایسه کوبرنتیس و داکر

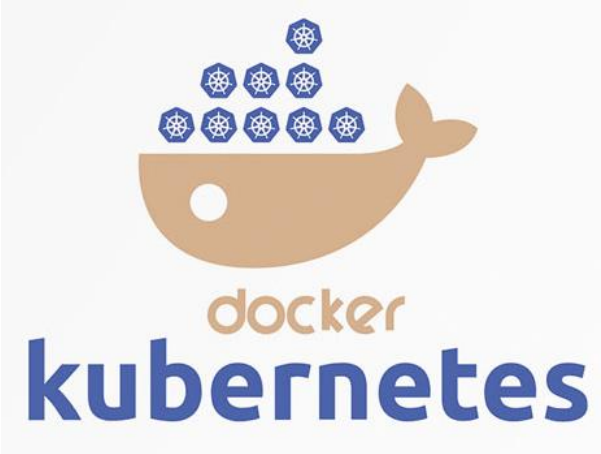
داکر (Docker) هم ابزاری متن‌باز برای مدیریت کانتینرهاست و طرفداران زیادی دارد. نسخه متن‌باز داکر که Community Edition نیز نامیده می‌شود، کاملاً رایگان است؛ اما درکنار آن نسخه‌ای پولی به نام Enterprise Edition هم ارائه شده است که امکانات اضافه‌ای برای مدیریت کانتینرها و پشتیبانی دارد.



قابلیت‌های این دو ابزار تا حدودی متفاوت است؛ اما درمجموع، داکر ازنظر امکانات یک رده پایین‌تر از Kubernetes قرار دارد. بسیاری Kubernetes را به‌دلیل ویژگی ماژولار و انعطاف‌پذیرتری بیشتر و سازگاری بیشتر با نیازهای سرویس‌دهنده‌های وب ترجیح می‌دهند.

داکر ازنظر امنیت در سطح بهتری قرار دارد و کوبرنتیس از اکوسیستم متنوع‌تری برای مدیریت کانتینرها برخوردار است. برای مثال، کوبرنتیس می‌تواند از داکر برای افزایش قابلیت‌های مدیریتی و به‌عنوان ران‌تایم کانتینر استفاده کند.

وقتی Kubernetes یک پاد را برای گره زمان‌بندی می‌کند، سرویس Kubelet در آن گره فرمان راه‌اندازی کانتینرهای خاصی را به داکر می‌دهد. سپس، Kubelet به‌صورت پیوسته وضعیت آن کانتینرها را از داکر دریافت و در سطح کنترل جمع‌آوری می‌کند. داکر کانتینرها را به درون نود هدایت می‌کند و آن‌ها را شروع و خاتمه می‌دهد. بدین‌ترتیب، استفاده از کوبرنتیز با داکر بخشی از وظایف ادمین را به سیستم محول می‌کند.



به‌طور خلاصه، Docker در درجه اول بر ایجاد و مدیریت کانتینرهای فردی متمرکز است و بسته‌بندی و توزیع برنامه‌ها را آسان‌تر می‌کند. از سوی دیگر، Kubernetes بر هماهنگ‌سازی و مدیریت برنامه‌های کاربردی کانتینری در مقیاس، خودکارسازی وظایفی مانند زمان‌بندی، مقیاس بندی و نظارت بر کانتینرها در یک خوشه تمرکز دارد.

در عمل، Docker و Kubernetes می‌توانند با هم کار کنند. Docker اغلب برای ساخت ایمیج کانتینر استفاده می‌شود که سپس می‌توان با استفاده از Kubernetes استقرار و مدیریت کرد. Kubernetes می‌تواند از کانتینرهای Docker به عنوان محیط زمان اجرا برای برنامه‌ها استفاده کند، در حالی که قابلیت‌های هماهنگ‌سازی و مدیریت لازم را برای اطمینان از استقرار کارآمد و مقیاس‌پذیر کانتینرها فراهم می‌کند.

مؤلفه‌های کوبرنتیس

اکثر استقرارهای کارآمد Kubernetes روی زیرساخت‌های مجازی معمولی اجرا و تعداد فزاینده‌ای نیز روی سرورهای فیزیکی ساده پیاده می‌شوند. کوبرنتیس به‌عنوان ابزار مدیریت استقرار و چرخه عمر اپ‌های کانتینری کاربرد دارد و ابزارهای جداگانه‌ای نیز برای مدیریت منابع زیرساخت به‌کار می‌روند.

پس از استقرار Kubernetes، یک کلاستر خواهید داشت که شامل مجموعه‌ای از ماشین‌های کارگر به نام گره است. این گره‌ها اپلیکیشن‌های کانتینری را اجرا می‌کنند و هر کلاستر حداقل یک گره دارد. گره کارگر میزبان پادهاست. این پادها اجزای تشکیل‌دهنده بار کاری اپلیکیشن‌ها هستند. مدیریت گره‌های کارگر و پادها در

کلاستر برعهده سطح کنترل قرار دارد. درادامه، اجزای مختلف موردنیاز برای داشتن یک کلاستر کامل و کارآمد را معرفی می‌کنیم.

۱. اجزای سطح کنترل

اجزای سطح کنترل تصمیمات کلی مثل زمان‌بندی کلاستر یا شناسایی و پاسخ به رویدادهای کلاستر را اتخاذ می‌کنند؛ مثلاً شروع پاد جدید درصورت ناقص‌بودن کپی استقرار. اجزای سطح کنترل را می‌توان روی هر ماشینی در کلاستر اجرا کرد؛ اما برای سهولت بیشتر معمولاً تمام اجزا در یک ماشین و کانتینرهای کاربر در ماشین دیگری اجرا می‌شوند.

۲. kube-apiserver

سرور API یکی از اجزای سطح کنترل است که API کوبرنتیس را برای ارتباط با سیستم منتشر می‌کند. این سرور برای سطح کنترل کوبرنتیز فرانت‌اند محسوب می‌شود. kube-apiserver پیاده‌سازی اصلی سرور API کوبرنتیز است که برای مقیاس‌بندی افقی طراحی شده است. به‌عبارت‌دیگر، با استقرار نمونه‌های بیشتر مقیاس آن هم بزرگ‌تر می‌شود. با اجرای چند نمونه از kube-apiserver، می‌توانید ترافیک را متعادل کنید.

۳. ETCD

etcd مؤلفه ذخیره مقادیر مهم با قابلیت دسترسی درخورتوجه است که بین گره‌های مختلف توزیع می‌شود. وظیفه این مؤلفه ذخیره اطلاعات پیکربندی است و تنها ازطریق API سرور دردسترس قرار دارد؛ چراکه ممکن است حاوی داده‌های حساس باشد.

۴. kube–scheduler

این مؤلفه پادهای تازه‌ایجادشده و بدون گره را پیدا می‌کند و گرهی را برای اجرا به آن‌ها اختصاص می‌دهد. معیارهای مهم برای زمان‌بندی تصمیمات این مؤلفه شامل الزامات منابع فردی و جمعی، محدودیت‌های سخت‌افزاری و نرم‌افزاری، محل داده‌ها، تداخل بارهای کاری و ضرب‌الأجل است.

۵. kube-controller-manager

این مؤلفه پردازه‌های کنترلر را اجرا می‌کند. ازنظر منطقی، هر کنترلر یک پردازه جداگانه است؛ اما برای کاهش پیچیدگی همه آن‌ها در یک باینری کامپایل و در یک پردازه اجرا می‌شوند. برخی از انواع این کنترلرها عبارت‌اند از:

- کنترلر گره: مسئول تشخیص و پاسخ‌دهی درصورت داون‌شدن گره
- کنترلر EndpointSlice: برای ایجاد ارتباط بین سرویس‌ها و پادها، اشیای EndpointSlice را جمع‌آوری می‌کند.
- کنترلر ServiceAccount: برای فضا نام جدید ServiceAccounts پیش‌فرض را ایجاد می‌کند.

۶. cloud-controller-manager

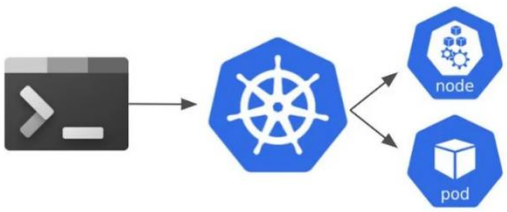
این مؤلفه امکان برقراری ارتباط بین کلاستر و API ارائه‌کننده سرویس ابری را برایتان فراهم می‌کند. دیگر کاربرد آن جداسازی مؤلفه‌های مرتبط با پلتفرم ابری از مؤلفه‌های لینک‌شده به کلاستر است. cloud-controller-manager تنها آن دسته از کنترلرهایی را اجرا می‌کند که مختص سرویس‌دهنده ابری شما هستند. اگر Kubernetes را روی سیستم محلی یا در محیط آموزشی اجرا کنید، کلاستر مدیر کنترلر ابر نخواهد داشت.

این مؤلفه هم مثل kube-controller چندین حلقه کنترل مجزا را در یک باینری ترکیب می‌کند تا به‌عنوان یک پردازه اجرا شوند. این کنترلرها به سرویس‌دهنده ابری وابسته هستند:

- کنترلر گره: بررسی سرویس‌دهنده برای اطمینان از حذف یک گره درصورت پاسخ‌گونبودن به درخواست‌ها
- کنترلر مسیر: تنظیم مسیرها در زیرساخت ابری
- کنترلر سرویس: ایجاد و به‌روزسانی و حذف متعادل‌کننده‌های بار سرویس‌دهنده

۷. پاد

پاد به‌عنوان کوچک‌ترین بخش کوبرنتیس شامل کانتینرهای مستقر در یک گره است و نقش بسیار مهمی در مدیریت اپلیکیشن کانتینری ایفا می‌کند. بین کانتینرهای یک پاد آدرس IPC،IP، نام میزبان و دیگر منابع مشترک است. کانتینرهای درون یک پاد از آدرس IP مختلفی درمقایسه‌با کانتینر پادهای دیگر استفاده می‌کنند. بدون وجود پاد، اجزای اپلیکیشن یا سرویس باید روی ماشین مجازی یکسانی اجرا شوند؛ اما پاد شبکه و حافظه را از زیرساخت پایه مستقل می‌سازد تا پروژه روی زیرساخت‌های مختلف اجراشدنی باشد.



در کوبرنتیز، کاربر می‌تواند شخصاً پاد ایجاد کند یا آن را به کنترلر بسپارد تا با بازدهی بیشتر این کار را انجام دهد. دلیل این بازدهی چشمگیر آن است که سرویس زمان‌بندی کوبرنتیز براساس تنظیمات و محل و محتویات پاد، بهترین گره را برایش پیدا می‌کند. در Kubernetes، ارتباط بین پادها به‌سادگی به‌واسطه سرویس‌ها شکل می‌گیرد. کانتینرها یکدیگر را ازطریق میزبان محلی پیدا و ازطریق استانداردهای مختلف مثل POSIX یا SystemV ارتباط برقرار می‌کنند.

در کوبرنتیس، می‌توانید برای هر پاد ماهیت و منابع مشخصی تعریف کنید. این مزیت به سیستم اجازه می‌دهد پادهای مرتبط به یکدیگر را در گره‌ای مناسب قرار دهد. برای مثال، اگر فرانت‌اند یک برنامه در پاد A یک گره قرار داشته باشد، قسمت بک‌اند را می‌توان در پاد B همان گره قرار داد تا به‌سادگی بین آن‌ها ارتباط برقرار شود.

پادها تا وقتی‌که کاربردی باشند یا حذف نشوند, در گره مشخص‌شده باقی خواهند ماند. با خاموش‌شدن هر گره, پادهای مربوط به آن براساس زمان‌بندی حذف خواهند شد تا منابع سیستم به‌هدر نرود.

۸. کنترلر تکرار

این کنترلر مشخص می‌کند که چه تعداد کپی مشابه از یک پاد باید در کلاستر اجرا شوند.

۹. سرویس

پراکسی سرویس کوبرنتیس به‌طورخودکار درخواست‌ها را به پاد مدنظر ارسال می‌کند. این مؤلفه حتی اگر پاد جایگزین یا جابه‌جا شده باشد, آن را پیدا می‌کند.

۱۰. Namespace

این سرویس راهی برای جداسازی فضاها در کوبرنتیس است. با دسته‌بندی اپ‌ها و منابع در Namespaceهای مجزا, می‌توان آن‌ها را به‌راحتی سازمان‌دهی کرد. برای مثال, اگر چند اپلیکیشن را در یک فضا قرار دهیم, باید نام‌های متفاوتی داشته باشند؛ اما با قراردادن هرکدام در یک Namespace جداگانه, می‌توانیم اپ‌هایی با نام یکسان را در فضاهای متفاوت داشته باشیم.

Namespace کلاسترها را به واحدهای هوشمند و ایزوله تبدیل می‌کند تا داده‌ها بهتر و آسان‌تر مدیریت شوند. Kubernetes به‌طورخودکار چهار Namespace را به منابعی اختصاص می‌دهد که مستقر شده‌اند؛ اما هنوز تعیین نشده‌اند. این‌ها عبارت‌اند از:

- منابع مستقر
- پیکربندی سیستم
- کاربری سیستم
- اشیای اجاره‌ای

تعداد Namespaceها محدودیتی ندارند و استفاده اصولی از آن‌ها در تفکیک بخش‌های تست و توسعه کمک زیادی به تیم پروژه می‌کند. برای مثال، اپ‌های کوچک به Namespace نیازی چندانی ندارند؛ اما برای میکروسرویس‌های متعدد می‌توانید از آن‌ها بهره ببرید.

مؤلفه‌های گره یا نود

۱. Kubelet

این سرویس روی گره‌ها اجرا می‌شود و سلامت و عملکرد صحیح هر کانتینر را به سرور اصلی در یک کلاستر گزارش می‌دهد. Kubelet گره‌ها را ازطریق سرورهای API پیدا می‌کند و از اجرای صحیح همه کانتینرهای یک سرور مطمئن می‌شود. Kubelet کانتینرهایی را که Kubernetes ایجاد نکرده باشد، مدیریت نمی‌کند.

۲. kube-proxy

kube-proxy یکی از پراکسی‌های شبکه است که در کلاستر روی هر گره اجرا می‌شود و بخشی از سرویس کوبرنتیس را پیاده‌سازی می‌کند. این مؤلفه روی اجرای صحیح قواعد شبکه در گره‌ها نظارت می‌کند تا ارتباط بین کلاسترهای داخل یا خارج از شبکه برقرار بماند. درصورتی‌که لایه فیلتر پاکت سیستم‌عامل دردسترس باشد، kube-proxy از همان استفاده می‌کند؛ وگرنه خودش ترافیک را می‌فرستد.

۳. ران‌تایم کانتینر

ران‌تایم کانتینر (Container Runtime) نرم‌افزاری برای اجرای کانتینرهاست. این ابزار درواقع بخشی از موتور کانتینر به‌حساب می‌آید که با کرنل درارتباط است. کوبرنتیز از ران‌تایم‌های کانتینر مثل containerd و CRI-O و دیگر پیاده‌سازی CRI کوبرنتیس پشتیبانی می‌کند. CRI-O نوعی روش پیاده‌سازی CRI است که به کوبرنتیس اجازه می‌دهد تا از هر OCI برای اجرا استفاده کند. متداول‌ترین ران‌تایم runC است؛ اما دیگر ران‌تایم‌های کانتینر مانند crun و railcar و Kata Containers نیز استفاده می‌شوند.

چالش‌های استفاده از کوبرنیتز چیست؟

مسئله مهم دیگری که در رابطه با اینکه کوبرنتیز چیست اهمیت دارد چالش‌های آن است. استفاده از Kubernetes می‌تواند چالش‌هایی از جمله پیچیدگی و منحنی یادگیری شیب‌دار آن را ایجاد کند. به ابزارهای اضافی برای کارهایی مانند ورود به سیستم، نظارت و پیکربندی نیاز دارد.

در حالی که Kubernetes بسیار توسعه‌پذیر است و موارد استفاده مختلف را پشتیبانی می‌کند، کانتینرها را مستقر می‌کند، نه کد منبع و ممکن است بهترین راه‌حل برای هر بار کاری نباشد. با این حال، اکوسیستم Kubernetes طیف وسیعی از ابزارهای بومی ابری را برای رسیدگی به مسائل مربوط به حجم کاری خاص ارائه می‌دهد و خدمات مدیریت شده Kubernetes ارائه شده توسط فروشندگان ابری می‌تواند پیچیدگی را کاهش داده و بهره‌وری توسعه‌دهندگان را افزایش دهد.

کوبرنیتز مدیریت شده چیست؟

Managed Kubernetes یا همان کوبرنتیز مدیریت شده سرویسی است که توسط ارائه‌دهندگان زیرساخت ابری مانند Oracle Cloud Infrastructure ارائه می‌شود که استقرار و مدیریت خوشه‌های Kubernetes را ساده می‌کند. با Kubernetes مدیریت شده، توسعه‌دهندگان می‌توانند به‌راحتی برنامه‌های کاربردی خود را در فضای ابری بسازند، مستقر کنند و مدیریت کنند.

با استفاده از یک سرویس مدیریت‌شده Kubernetes مانند موتور کانتینر زیرساخت ابری Oracle برای Kubernetes، توسعه‌دهندگان می‌توانند منابع محاسباتی مورد نیاز را برای برنامه‌های کاربردی خود و تدارکات خدمات مشخص کرده و زیرساخت‌های زیربنایی را در محیط ابری مدیریت کنند.

خدمات مدیریت شده Kubernetes مزایایی مانند در دسترس بودن بالا، کنترل، امنیت و عملکرد قابل پیش‌بینی را ارائه می‌دهد. آن‌ها هم از ماشین‌های فلزی و هم ماشین‌های مجازی به عنوان گره‌هایی برای اجرای خوشه‌های Kubernetes پشتیبانی می‌کنند. این خدمات توسط بنیاد محاسبات بومی ابری (CNCF) مطابقت دارند و از سازگاری با اکوسیستم گسترده‌تر Kubernetes اطمینان حاصل می‌کنند.

با استفاده از سرویس مدیریت‌شده Kubernetes، توسعه‌دهندگان می‌توانند از راحتی و سهولت استفاده ارائه‌شده توسط ارائه‌دهنده خدمات بهره ببرند، در حالی که با آخرین به‌روزرسانی‌های Kubernetes به‌روز می‌مانند و سازگاری با اکوسیستم CNCF را حفظ می‌کنند، بدون نیاز به تلاش اضافی یا کار دستی.

KubeCon چیست؟

اصطلاح و مفهوم مهم دیگری که در رابطه با اینکه کوبرنتیز چیست اهمیت دارد مفهوم KubeCon است. KubeCon یک کنفرانس سالانه بوده که به جامعه Kubernetes اختصاص داده شده است. این برنامه توسعه‌دهندگان، کاربران و علاقه‌مندان به Kubernetes و فناوری‌های بومی ابری را گرد هم می‌آورد. این رویداد از زمان آغاز به کار خود در سال ۲۰۱۵ رشد قابل‌توجهی داشته است و هرساله تعداد زیادی از شرکت‌کنندگان را به خود جذب می‌کند.

KubeCon به عنوان یک پلت فرم برای یادگیری، همکاری و به اشتراک‌گذاری تجربیات مربوط به Kubernetes و اکوسیستم کلود بومی ابری عمل می‌کند. این آخرین پیشرفت‌ها، بهترین شیوه‌ها و موارد استفاده در دنیای واقعی را به نمایش می‌گذارد و به رویدادی کلیدی برای جامعه بومی ابری برای جمع‌آوری، شبکه‌سازی و به‌روز ماندن با چشم‌انداز به‌سرعت در حال تکامل Kubernetes تبدیل شده است.

جمع‌بندی

Kubernetes پلتفرمی متن‌باز و توسعه‌پذیر و پرتابل برای پیاده‌سازی و مدیریت برنامه‌های کانتینری است. این پلتفرم در سال ۲۰۱۴ منتشر شده و حاصل تجربه ۱۵ساله متخصصان گوگل درزمینه پردازش و اجرای اپ و سرویس در مقیاس بسیار بزرگ است.

کانتینر به‌عنوان مؤلفه اصلی کوبرنتیس پکیجی است که فایل‌های سیستمی و کتابخانه و کدهای یک پروژه را در خود دارد و به اعضای تیم اجازه می‌دهد پروژه را در محیط‌های مختلف سرور مجازی لینوکس و سرور فیزیکی یا ابری و حتی هیبرید اجرا و تست کنند.

سؤالات متداول

۱. کوبرنتیس چیست؟

کوبرنتیس را گوگل معرفی کرده و پلتفرمی متن‌باز و پرتابل برای استقرار و مدیریت و مقیاس‌بندی آسان برنامه‌های کانتینری است.

۲. کوبرنتیس بهتر است یا داکر؟

داکر کاربری آسان‌تری از کوبرنتیس دارد و برای سازمان‌های کوچک مناسب‌تر است. درمقابل, Kubernetes قابلیت‌های بیشتری از داکر دارد.

۳. کاربردهای کوبرنتیس چیست؟

کاربردهای کوبرنتیس عبارت‌اند از: استفاده بهینه از سخت‌افزار, توسعه سریع نرم‌افزار, بهینه‌سازی و اصلاح آسان محصول, افزایش آنی منابع, تست و توسعه پروژه و مستقل‌بودن از زیرساخت.

منابع

۱. پارس پک (سانیا عبدی پور) [+]
۲. ایران سرور [+]
۳. سون لرن [+]
۴. مکتب خونه [+]
۵. Vmware [+]
۶. Oracle [+]