Spring Cloud

Milad Barzideh

فهرست مطالب

۵	میکروسرویسها، ابر و چیزهای دیگر	1
۵	 ۱.۱ میکروسرویس چیست؟	
٧	 ۲.۱ محاسبات ابری چیست؟	
٧	 ٣.١ چرا ابر؟	

فهرست مطالب

فصل ۱

میکروسرویسها، ابر و چیزهای دیگر

میکروسرویس یا ریزخدمت کی از اصطلاحاتیست که این روزها در دنیای نرمافزار رایج شده است. میکروسرویسها، سرویسهایی توزیعشده و مجزا هستند که هر کدام از آنها وظیفهی کوچک و مشخصی را انجام میدهند. در این نوشتار، پس از معرفی میکرسرویسها و ذکر مشخصههای آن، با استفاده از فریمور کهای Spring Boot و Spring Cloud به صورت عملی، معماری میکروسرویسها را بررسی خواهیم کرد.

1.۱ میکروسرویس چیست؟

قبل از مطرح شدن میکروسرویسها بیشتر برنامههای تحتوب با استفاده از معماری یکپارچه ^۲ ساخته میشدند. در معماری یکپارچه، هر برنامه به عنوان یک محصول نرمافزاری قابل اجرا در نظر گرفته میشود و رابط کابری، منطق برنامه و دسترسی به پایگاه داده، همه در یک برنامهی کاربردی قرار گرفته میشود.

هر برنامه به عنوان یک واحد کاری در نظر گرفته می شود که معمولا چندین تیم توسعه روی بخشهای مختلف آن کار می کنند. یکی از مشکلاتی که در این حالت پیش می آید این است که با افزایش اندازه و پیچیدگی برنامه، هزینههای ارتباطی و هماهنگی تیمها، بیشتر و سخت تر می شود. برای مثال با هر تغییری که تیمها ایجاد می کنند، کل برنامه دوباره باید تست و مستقر "شود.

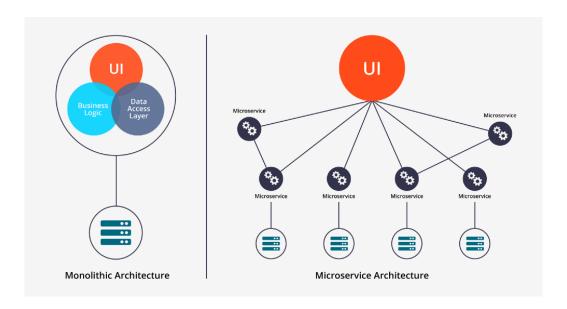
مفهوم میکروسرویس تقریبا از سال ۲۰۱۴ و در پاسخ به چالشهایی که برای مقیاسپذیر کردن برنامههای یکپارچه وجود داشت، وارد جامعهی نرمافزار شد. همانطور که گفته شد یک میکروسرویس، کوچک، مجزا و توزیعشده است. بنابراین این امکان را فراهم میکند که برنامههای بزرگ را به بخشهای کوچکتری تقسیم کرد تا مدیریت و توسعه ی آنها آسان تر داده شوند. مهم ترین نکته ای که باید به آن توجه داشته باشید، تجزیه و جداسازی عملکردهای برنامه است به صورتی که کاملا مستقل از همدیگر باشند.

همان طور که شکل ۱.۱ نشان می دهد هر تیم می تواند کدها و زیرساخت مربوط به سرویس خود را داشته باشد و آنها را به صورت مستقل نسبت به سرویسهای دیگر توسعه و ارزیابی کند.

[\]Microservice

⁷Monolithic

[&]quot;Redeployed



شکل ۱.۱: معماری یکپارچه در مقابل معماری میکروسرویس

معماری میکروسرویس ویژگیهای زیر را دارد:

- * منطق برنامه به قسمتهای کوچکتر شکسته می شود که هر قسمت مسئلهی مشخص و تعریف شده ای را حل می کند.
- * هر قسمت محدوده ی مسئولیت مشخصی دارد و مستقل از سایر بخشها توسعه و استقرار می یابد. همچنین میکروسرویسها باید قابلیت استفاده ی مجدد در برنامههای دیگر را داشته باشند.
- * میکروسرویسها با استفاده از قواعد مشخص و بکارگیری پروتکلهای ارتباطی سبکوزن مانند HTTP و JSON دادهها را مبادله می کنند.
- * پیادهسازی فنی هر میکروسرویس بیاهمیت است؛ زیرا برنامهها با پروتکلهای مستقل از تکنولوژی ٔ با هم ارتباط برقرار میکنند. این یعنی برنامهای که با معماری میکروسرویس ساخته شده می تواند از تکنولوژیها و زبانهای مختلفی استفاده کند.
- * میکروسرویسها برای سازمانها این امکان را فراهم میکنند که تیمهای توسعه ی کوچک با وظایف کاری مشخص داشته باشند.

^{*}Technology-neutral protocol

۲.۱ محاسبات ابری چیست؟

٣.١ چرا ابر؟

به عنوان یک توسعه دهنده ی میکروسرویس، دیر یا زود باید تصمیم بگیرید سرویس تان را در یکی از جاهای زیر مستقر کنید:

- * سرور فیزیکی: سازمانهای کمی این کار را انجام میدهند؛ زیرا سرورهای فیزیکی محدودیت ایجاد می کنند. برای مثال شما نمی توانید ظرفیت سرور فیزیکی را سریعا افزایش دهید یا میکروسرویس را به دلیل هزینههایی زیادی که در پی دارد روی چندین سرور فیزیکی مستقر کنید.
- * ماشین مجازی: یکی از مهمترین فواید میکروسرویسها، توانایی آنها در افزودن یا کاهش نمونههای یک سرویس است (مقیاسپذیری). یک میکروسرویس را میتوان در یک ماشین مجازی قرار داد و به آسانی چندین نمونه از آن را روی زیرساختهای موجود مستقر کرد.
- * ظرف مجازی ^۵: به جای استقرار میکروسرویس روی یک ماشین مجازی کامل، بسیاری از توسعه دهندگان سرویسهای خود را به صورت ظرفهای داکر ^۶ در محیط ابری مستقر می کنند.

تفاوت داکر با ماشین مجازی

مزیت مهم میکروسرویسهای تحت ابر انعطاف پذیری بالای آنهاست. ارائه دهندگان محیط ابری اجازه می دهند در کمتر از چند دقیقه یک ماشین مجازی یا یک ظرف جدید را اضافه یا حذف کنید. همچنین برنامهها مقاومتر هستند، برای مثال اگر یکی از میکروسرویسها از کار بیفتد می توان از نمونههای دیگر آن میکروسرویس استفاده کرد و برنامه را بالا نگه داشت.

^aVirtual container

Docker

فصل ۲ کنترل پیکربندیها