



معماری میکروسرویس (Microservice Architecture) به واسطه مزیت‌هایی مانند مقیاس‌پذیری، پایداری و انعطاف‌پذیری بالا، جهان توسعه نرم‌افزار را دگرگون کرده است. در طول سال‌های اخیر، بسیاری از سازمان‌ها از **معماری مونولیتیک** به معماری میکروسرویس مهاجرت کرده‌اند؛ البته این اقدام باید با احتیاط و برنامه‌ریزی دقیق انجام شود تا نقاط قوت آن به‌درستی در دسترس شما قرار گیرند. پیش‌تر، یک **آموزش جامع میکروسرویس** در اختیار شما قرار داده‌ایم که با مطالعه آن می‌توانید بایدها و نبایدها و مسیر شغلی آن را درک کنید. اکنون در این مطلب، به بررسی بهترین روش‌های پیاده‌سازی معماری میکروسرویس و Best Practice های آن پرداخته می‌شود.

### معماری میکروسرویس چیست؟

میکروسرویس یک معماری نرم‌افزار است که در آن، اپلیکیشن به مجموعه‌ای از سرویس‌های مستقل و کوچک تقسیم و ایجاد می‌شود. در این معماری، هر میکروسرویس عملکرد مشخص و تعریف‌شده‌ای دارد و برای برقراری ارتباط با سایر سرویس‌ها، از API های سبک (Lightweight) استفاده می‌کند. از مهم‌ترین مشخصه‌های معماری میکروسرویس، می‌توان به استقلال عملیاتی هر میکروسرویس، **اتصالات سست** (Loose Coupling) سرویس‌ها، امکان مقیاس‌پذیری و سازگاری آن با آخرین تکنولوژی‌های روز دنیای فناوری اشاره کرد. ما در **مقاله میکروسرویس چیست**، به بررسی کامل آن پرداخته‌ایم که پیشنهاد می‌کنیم حتماً آن را مطالعه نمایید.



## مزایای معماری میکروسرویس

مزیت های معماری میکروسرویس در ادامه لیست شده اند:

- **انعطاف پذیری:** در صورت نیاز، امکان سازگاری و تکامل با تغییرات را دارد.
- **پایداری:** اگر یکی از سرویس ها در معماری میکروسرویس به مشکل بخورد، مشکلی در سایر سرویس ها به وجود نمی آید.
- **مقیاس پذیری:** شما می توانید با افزودن سرویس هایی به معماری میکروسرویس، آن را به صورت افقی Scale کنید.
- **توسعه سریع:** افراد تیم این امکان را دارند که به صورت مستقل بر روی بخش های جداگانه ای از سرویس ها کار کنند.
- **بهبود نگهداری کد:** درک و آپدیت Codebase های کوچک آسان تر است.

علاقه مندان می توانند برای آشنایی بیشتر با جزئیات و نقاط مثبت این معماری، [مقاله مزایای معماری میکروسرویس](#) را مطالعه کنند.

## معایب معماری میکروسرویس

مزیت های میکروسرویس غیرقابل انکار است؛ با این وجود، پیاده سازی معماری میکروسرویس کاستی هایی دارد که در ادامه به آن اشاره خواهد شد.

- احتمال بروز پیچیدگی در **مدیریت ارتباط بین سرویس ها**
- توجه به سازگاری داده ها روی سرویس های مختلف
- نیاز به رویکردهای مختلفی برای تست و اشکال زدایی

با این مقدمه، در ادامه به بررسی بهترین روش های پیاده سازی معماری میکروسرویس می پردازیم.

## ۵ نکته مهم برای پیاده سازی معماری میکروسرویس

با دنبال کردن اصول و بهترین روش های پیاده سازی معماری میکروسرویس ، می‌توانید از ویژگی‌های قدرتمند این معماری بهره‌مند شوید. این Best Practice ها ، در ادامه شرح داده می‌شوند.



### ۱- اصول طراحی: ایجاد پایه قدرتمند

زمانی که از اصول طراحی معماری میکروسرویس صحبت می‌شود، رعایت کردن موارد زیر، کمک‌کننده خواهند بود:

- **اصول SOLID:** استفاده از **اصول SOLID** امکان تست، نگهداری و Loose Coupling سرویس‌ها در معماری میکروسرویس را تضمین می‌کنند.
- **Bounded Context ها:** تعیین مرزبندی‌های شفاف براساس دامنه، مواردی همچون کاهش پیچیدگی، امکان توسعه و استقرار مستقل را ترویج می‌دهد. برای آشنایی بیشتر، به **مقاله Bounded Context در توسعه نرم افزار** رجوع کنید.
- **API Gateway:** می‌توان به‌واسطه API Gateway متمرکز، ترافیک را مدیریت کرد، سیاست‌های امنیتی مختلفی را اعمال کرده و تعاملات کلاینت را تسهیل بخشید. بدین طریق، یک Access Point تحت کنترل و انتزاعی ارائه خواهد شد. در **مقاله میکروسرویس: API Gateway چیست؟** این موضوع را به‌طور کامل شرح داده‌ایم که به شما توصیه می‌کنیم حتماً آن را مطالعه نمایید.
- **معماری رویدادمحور (Event-Driven Architecture):** در شرایطی که لازم باشد ارتباط سرویس‌ها به‌صورت سست و مقیاس‌پذیر باشد، این رویکرد، به‌ویژه در تعاملات غیرهمگام (Asynchronous Interactions)، مناسب است.

- **CQRS (Command Query Responsibility Segregation)**: جداسازی مدل‌های Read و Write به بهبود کارایی و سازگاری منجر می‌شود. این موضوع به صورت ویژه برای اپلیکیشن‌هایی مناسب است که امکان پردازش داده‌های وسیعی را دارند.
- **API Versioning**: برای رسیدگی به تغییرات گسترده، پیشنهاد می‌شود که از استراتژی‌های همچون URL Path Versioning یا Header-Based Versioning بهره‌مند شوید.

## ۲- بهترین روش های توسعه: ساخت سرویس های کارآمد

در گام دوم از پیاده سازی معماری میکروسرویس ، به بررسی موارد مهم در توسعه آن پرداخته می‌شود:

- **انتخاب تکنولوژی**: انتخاب تکنولوژی، یکی از بنیادی‌ترین اقدامات لازم برای پیاده سازی معماری میکروسرویس محسوب می‌شود و لازم است با دقت و احتیاط انجام شود. تکنولوژی انتخابی باید با نیازمندی‌های پروژه، تخصص تیم فنی و نگهداری درازمدت (Long-term Maintenance) اپلیکیشن هم‌تراز باشد. در این مسیر با **انواع زبان های برنامه نویسی** ، مانند سی شارپ و همچنین فریمورک‌های مختلف کار خواهید کرد.
- **میکرو فرانت اند ها (Micro Frontends)**: اگر ال پیچیده باشد، می‌توان از Micro Frontend ها به هدف توسعه ماژولار، تسهیل نگهداری و امکان استفاده مجدد بهره‌مند شد. به‌عنوان مثال، شرکت‌های همچون Airbnb ، Spotify یا توییتر از میکروفرانت‌اندها برای توسعه ال استفاده می‌کنند.
- **تست نویسی جامع**: با پیاده‌سازی انواع استراتژی تست نویسی، شامل **Unit Testing** و **Testing Integration** و **End-To-End Testing** از کیفیت سرویس اطمینان پیدا کنید.
- **روش‌های CI/CD**: به منظور کاهش مداخله‌های Manual، عرضه سریع و آسان، به خودکارسازی استقرارها از طریق ابزارهای کارآمد بپردازید.

## ۳- استراتژی های استقرار معماری میکروسرویس: تسهیل انتشار

در زمینه استقرار معماری میکروسرویس، می‌توان استراتژی‌های زیر را به کار برد:

- **کانتینر سازی**: استقرار سرویس‌ها در کانتینرهای ایزوله (داکر و کورننتیس)، مزیت‌هایی مانند استفاده بهینه از منابع، قابل حمل بودن (Portability) و بهبود مقیاس‌پذیری را به همراه دارد.
- **میکروسرویس‌های بدون سرور**: برای رسیدن به مقیاس‌پذیری بهبودیافته و مقرون‌به‌صرفه، امکان بررسی گزینه‌هایی همچون AWS Lambda و Azure وجود دارد. این موضوع به طور ویژه برای آن دسته از لودهای کاری مناسب است که مبتنی بر رویداد (Event-Driven) هستند.
- **زیرساخت به‌عنوان کد (IaC)**: خودکارسازی فرآیند تهیه و پیکربندی زیرساخت با استفاده از ابزارهایی مانند Terraform یا Ansible ممکن است. بدین طریق، تطبیق‌پذیری، امکان استفاده مجدد و مدیریت روی محیط‌های مختلف تسهیل می‌یابند.
- **استقرار Blue-Green**: همزمان با نسخه‌های موجود، به استقرار نسخه‌های جدید بپردازید تا به تدریج ترافیک شیفت بیابد.
- **استقرار Canary**: می‌توان نسخه‌های جدید را ابتدا برای زیرمجموعه کوچک‌تری از کاربران استقرار کرد تا پیش از هر موضوعی، از پایداری اطمینان پیدا کرده و بازخوردهای افراد را دریافت کنید.

#### ۴- ضرورت های مدیریت معماری میکروسرویس : مانیتورینگ و نگهداری

برای نظارت و نگهداری میکروسرویس ، اقدامات زیر مفید خواهند بود.

- **ابزارهای مانیتورینگ:** نظارت مستمر روی کارایی و سلامت سرویس ها، از طریق **پرومتئوس** یا Grafana مطلوب خواهد بود. به این ترتیب، با پیگیری معیارهای کلیدی مانند تأخیر (Latency)، نرخ خطا و منابع مورد استفاده، می توان به صورت فعالانه، مشکلات بالقوه را تشخیص داد.
- **لاگ گیری:** می توان با پیاده سازی شیوه های Logging، سطوح مختلفی از آن را برای به دست آوردن اطلاعات مرتبط به کار برد.
- **هشدار و نوتیفیکشن:** با راه اندازی هشدارهایی برای شرایط بحرانی، این امکان فراهم می شود که اعضای تیم به سرعت از آن ها با خبر شوند و به فکر راه چاره ای باشند.

#### ۵- ملاحظات امنیتی: محافظت از معماری میکروسرویس

با در نظر داشتن موارد زیر، به حفظ امنیت معماری میکروسرویس پردازید:

- **Authorization و Authentification:** با اعمال مکانیزم های احراز هویت و اعطای مجوز، از سرویس های معماری میکروسرویس خود حفاظت کنید. شما می توانید استانداردهایی مانند OAuth و OpenID را برای کنترل دسترسی کاربر پیاده سازی کنید.
- **رمزگذاری داده ها:** می توان داده های حساس را از طریق الگوریتم های رمزگذاری (Encryption) محافظت کرد. شایان ذکر است که ابزارهایی مانند TLS/SSL به منظور ایمن سازی کانال های ارتباطی مورد استفاده قرار می گیرند.
- **ایزوله سازی میکروسرویس:** سرویس ها را در محیط های ایزوله استقرار کنید تا ریسک های امنیتی کمینه شوند و تأثیرات مربوط به تخلف های احتمالی محدود شوند.
- **استفاده از اصول Zero Trust:** با پیاده سازی اصول Zero Trust، تمامی درخواست ها صحت سنجی می شوند و دسترسی های Privilege به صورت حداقلی اعطا خواهند شد.
- **بررسی مستمر نقاط آسیب پذیری:** اگر سرویس ها و کانتینرها به صورت مستمر از نظر آسیب پذیری اسکن شوند، می توان به واسطه Patch هایی مشکلات را پیگیری کرد.

#### بررسی آماری: رشد تصاعدی معماری میکروسرویس

معماری میکروسرویس در طول سال های اخیر به شهرت فراوانی رسیده است. این معماری نرم افزار، به دلیل چابکی، امکان مقیاس پذیری مستقل و بهبود قابل اکتفا بودن آن، توسط بسیاری از سازمان ها به کار می رود. داده های آماری گوناگونی، رشد نمایی معماری میکروسرویس را تأیید می کنند. بررسی **گزارش CNCF Survey 2023** نشان می دهد که میکروسرویس توسط ۸۴٪ از شرکت کنندگان این پرسشنامه استفاده می شود. ضمن اینکه ۶۲٪ از افراد آن را در محیط محصول (Production Environment) به کار می برند. علاوه بر این، با استناد به **داده های آماری گزارش RightScale State of the Cloud 2023** بیش از ۷۸٪ از سازمان ها از میکروسرویس استفاده می کنند.



### میکروسرویس: معماری آینده برای توسعه نرم افزار

رشد قابل توجه معماری میکروسرویس غیرقابل انکار است و به همین دلیل، بسیاری از سازمان‌های بزرگ در صد مهاجرت به آن هستند. با پیاده‌سازی معماری میکروسرویس می‌توان از چابکی، مقیاس‌پذیری و قابل اکتفا بودن آن در توسعه نرم افزار مدرن بهره‌مند شد. در این مقاله به بررسی بهترین روش های پیاده سازی معماری میکروسرویس و Best Practice ها پرداخته شد. برای آن که استفاده از این معماری موفقیت‌آمیز باشد، برنامه‌ریزی دقیق و انتخاب ابزارهای مناسب اهمیت فراوانی دارد.