در این ویدئو نگاهی به یک معماری نمونه خواهیم انداخت که برای داشتن درک مشترک از پروژه های کوچه تا بزرگ مناسب است.

در صورتی که شما پروژه های مختلف که با معماری مایکروسرویس پیاده سازی شده اند را بررسی نمایید متوجه این موضوع خواهید شد که آن ها با وجود تفاوت ها در هسته مرکزی یا Core بسیار به هم شبیه هستند.

خوب اجازه بدبد با مثال منظورم را واضح تر بیان کنم :

ما اینجا یک مثال از معماری مایکروسرویس داریم که شامل تعدای مایکروسرویس و تعدای پروژه Spring Cloud بسیار مهم که برای انجام عملیات مایکروسرویس ها در cloud ضروری هستند.

شما در دیاگرام تعدادی خط های فلش دار می بینید که منظورمان از آنها فقط یک ارتباط ساده نیست.

خوب اجازه بدید بریم یه سری به آنها بزنیم و بیشتر با جزئیات آنها آشنا بشیم تا شما درک بیشتری از این معماری پیدا کنید و بتونید یک پروژه شبیه به این را در پروژه خودتان پیاده سازی کنید

زمانی که پروژه خودمون را با مایکروسرویس پیاده سازی می کنیم سعی می کنیم که هر کار مربوط به کسب و کارمان را با استفاده از مایکروسرویس ها که هر کدام مسئولیت یک کار را بر عهده دارند پیاده سازی کنیم.

در این دیاگرام من طراحی Application خودم را با استفاده از روش مایکروسرویس آغاز کرده ام که هر مایکروسرویس وظیفه انجام یک کار و یا یک دامنه کاری را برعهده دارند.

بطور مثال من یک مایکروسرویس به نام Usersدارم که وظیفه آن می تواند ثبت کاربران، گرفتن جزئیات اطلاعات کاربران و بروزرسانی اطلاعات کابران در یک دیتابیس باشد.بنابراین موجودیت اصلی در این مورد یک کاربر است و این مایکروسرویس تنها با کاربر کار داردو عملیات دیگری را انجام نمی دهد و تمرکز آن بر روی جزئیات کاربران می باشد و دیتابیس خودش را نیز دارد.

در Application های مونولیتیک ما تنها یک دیتابیس داریم و اطلاعات مربوط به همه چیز را در آن دیتابیس قرار می دهیم و آن اطلاعات را در جداول متفاوتی قرار می دهیم.

در معماری مایکروسرویس به این نکته توجه کنید که هر یک از مایکروسرویس ها دیتابیس خودشان را دارند .بطور مثال در این دیاگرام مایکروسرویس Usersدیتابیس خودش را دارد و اطلاعات خودش را در آن دیتابیس ثبت و نگهداری می کند.

مایکروسرویس بعدی با نام Photos می باشد و مسئولیت آن نگهداری عکس هایی که کاربر ارسال می کند و برگرداندن دیتاهای آن عکس ها و انجام پردازش روی عکس ها می باشد.

مایکروسرویس های دیگری به نام های Album Microservice , order Microservice و Notification Microservice داریم که هر کدام مسئولیت های مربوط به خود را دارند.مثلا مسئولیت Notification Microservice می تواند برداشت notification ها از صف و ارسال آنها به ایمیل کاربران باشد.

در معماری مایکروسرویس یک مایکروسرویس جهت ارتباط با سایر مایکروسرویس ها از پروتکل HTTP استفاده می کنند.بطور مثال اگر مایکروسرویس Order بخواهد با مایکروسرویس Notification ارتباط برقرار کند یک HTTP request به سمت مایکروسرویس Notificationارسال می کند یا می تواند درخواست خود را در Queue، Produce کند و یکی از instance های مایکروسرویس Notification آن درخواست را consume و پردازش کند.

بصورت مشابه اگر مایکروسرویس usersبخواهد لیستی از عکس ها را از مایکروسرویس Photos درخواست کند نباید به دیتابیس مایکروسرویس Photos وصل شود پس برای این کار باید یک Http Request به مایکروسرویس Photosبزند و لیست عکس ها را دریافت کند.

مایکروسرویس بعدی که در این معماری داریم سرویس Discovery هستش که ما در مورد این مایکروسرویس در این ویدئو توضیحاتی را می دهیم و شما در ادامه با ایجاد و اجرای آن آشنا خواهید شد.

زمانی که یک Client Application می خواهد یک درخواست را به Users Microservice ارسال کند نیاز است که URL یا آدرس مایکروسرویس Usersرا بداند .درحالی که ما می توانیم چندین نمونه یا Instance از این مایکروسرویس را با آدرس ها و پورت های متفاوت در حال اجرا داشته باشیم و Client Application نمی تواند آدرس تمام امایکروسرویس ها را نگهداری کند .بنابراین ما از Discovery service استفاده می کنیم .

Discovery service به این شکل عمل می کند که زمانی که یک مایکروسرویس استارت می شود خود را در Discovery service رجیستر می کند و پس از آن Client Application درخواست خود را می تواند به Users Microservice از طریق API Gateway ارسال کند و API Gateway آن درخواست را بصورت مستقیم به Microservice مورد نظرارسال می کند چرا که آدرس همه آنها را از طریق Discovery Service می داند.

در این دیاگرام من تمام ارتباطات را بدلیل پیچیده شدن به هم متصل نکردم ولی تمام مایکروسرویس باید خود را در Discovery Service رجیستر کنند اگر قرار است که قابل دسترسی باشند.

خوب ما در این معماری API Gateway را داریم که تنها نقطه مرکزی و ورودی برای تمام HTTP Requestها ازClient Application ها هستش و از طریق Load Balancer هر HTTP Request به یکی از نمونه ها یا Instance های درحال اجرای مایکروسرویس مورد نظر ارسال می شود.بنابراین در یک معماری مایکروسرویس شما Load Balancer را خواهید داشت .

در ادامه شما یاد خواهید گرفت چطور API Gateway خودتان را ایجاد کنید و اینکه چگونه Load Balancer کار می کند.

آخریم مایکروسرویسی که در این معماری مایکروسرویس خواهیم داشت مایکروسرویس Config Server هست که ما در ادامه یاد می گیریم چطور آن را پیکربندی و اجرا کنیم .ما با استفاده از Configuration service می توانیم تنظیمات مربوط به مایکروسرویس های خود را که بصورت اشتراکی در مایکروسرویس های دیگر هم وجود دارند مانند تنظیمات مربوط به اتصال به Queue یا دیتابیس یا غیره ، بصورت متمرکز نگهداری نگهداری کنیم چرا که در صورت نیاز به انجام تغییر در آنها دیگر نیاز نباشد این تغییرات را در تمام مایکروسرویس ها انجام دهیم و مجبور باشیم تمام آنها را مجددا Restartکنیم.

با استفاده از Configuration Service ما می توانیم این تنظیمات را بصورت متمرکز نگهداری کنیم و در صورت نیاز آنها را بروزرسانی کنیم و آنها را به تمام مایکروسرویس های دیگر Push کنیم و نیازی به Restart کردن مایکروسرویس نداریم.

خوب این نمونه ای از معماری مایکروسوریس بود که برای شما توضیح دادم و در ادامه شما زمانی که بیشتر در مورد مایکروسرویس یاد می گیرید با ورژن های متفاوتی مواجه خواهید شد که همچنان به این معماری بسیار شبیه هستند.