এই ব্লগে আমরা Software বা Web Application ডেভেলপমেন্টের সবচেয়ে গুরুত্বপূর্ণ আর্কিটেকচার, Microservice নিয়ে সংক্ষেপে আলোচনা করব।

Microservices আসলে কি?

Microservice হল, একটি সিস্টেমের কম্পোনেন্টগুলোকে ছোট ছোট সতন্ত্র মডিউলে রান করা।। এখানে প্রতিটি সিস্টেম একটি সিঙ্গেল ইউনিট হিসেবে ডেভেলপ এবং ডেপ্লয় করা হয় । একটি ইউনিট বা সার্ভিস আরেকটি সার্ভিসের সাথে তাদের নিজস্ব API এর মাধ্যমে যোগাযোগ করবে। মাইক্রোসার্ভিস এর মাধ্যমে, ডেভেলপাররা একটি বড় App কে ভেঙ্গে অনেকগুলো ছোট ছোট App তৈরী করে, যেমন: ম্যাপ ভিউ , লোকেশন নির্নয় ইত্যাদির জন্য লোকেশন সার্ভিস, ফাইল আপলোড, রিসাইজ, সেভ, ডাউনলোড হ্যান্ডেলিং এর জন্য আরেকটি সার্ভিস কিংবা Facebook বা Twitter এ কিছু শেয়ার করার জন্য একটি ভিন্ন সার্ভিস তৈরী করা হয়। এবং এই সার্ভিসগুলো API এর মাধ্যমে একে অপরের সাথে সংযুক্ত থাকে।

Microservices এর বিভিন্ন ধরনের Criteria রয়েছে। যেমনঃ

- Codebase: এতে codebase আলাদা আলাদা থাকে।এর ফলে code maintain করা সহজ হয়। প্রত্যাকটি ছোট team কটি মাত্র Microservices এর কাজ করে এবং সেটি Documented করে রাখতে পারে।
- User Interface: User Interface একটি single application এর মতো এবং তার উপর ভিত্তি করে UI develop করতে হবে।এটার দুটি টেকনিক আছে।যেমনঃহয় Server side এ Microservice developer পেইজ বানায়ে ফ্রন্টে পাঠাবে অথবা ফ্রন্টে UI টিম থাকবে যারা সব ডেটা এক সাথে বানাবে।এতে Microservice এ UI নিয়ে চিন্তা না করলেও হবে।
- Data Storage: Microservices এ ডেটা স্টোরেজ গুলো আলাদা আলাদা থাকে, এতে আমরা বিভিন্ন requirement এ বিভিন্ন ধরনের database ব্যবহার করা করতে পারি।এর মাধ্যমে বিভিন্ন ধরনের database এর বিভিন্ন স্ট্র ফিচার গুলো আমরা ব্যবহার করতে পারি।যেমনঃ User এর জন্য LDAP ব্যবহার করতে পারি আর Order এর জন্য NoSQL আবার Product এর জন্য SQL database ব্যবহার করতে পারি।
- Data Synchronization: এ No distribution transaction ব্যবহার করা হয় না।এটা ব্যবহার করলে পুরো database update হয়ে যাবে। এর ফলে data synchronization স্লো হয়ে যাবে। এই জন্য এটা ব্যবহার করা হয় না। এতে data consistency রাখার জন্য Eventual consistency implement করতে হয়।এই data consistency রাখার জন্য তারা data change capture করে এবং Event sourcing এর মাধ্যমে তা জানিয়ে দেয়। এই কাজ টা করার জন্য বিভিন্ন ধরনের library ব্যবহার করা হয়। যেমনঃAkka, Kafka ইত্যাদি।

Microservices গুলো সাধারণত পরস্পরের সাথে কিভাবে Communicate করে?

Communicate সাধারণত দুইভাবে করা যায়। যেমনঃ

- Remote Procedure Invocation (RPC): RPC Request/Reply মেথড এ কাজ করে থাকে।এর বিভিন্ন ধরনের মেথড রয়েছে।যেমনঃ REST, SOAP, gRPC ইত্যাদি। JSON, REST এ একটি STP Request পাঠায় এবং JSON format এ কিছু ডেটা পাঠায়।
- Messaging: Broker অথবা Channel এর মাধ্যমে Microservice messaging করতে পারে।এখানে serviceএ গুলো subscribe করা থাকে। এর ফলে যদি কোন message কেউ channel এ publish দেয় তাহলে অন্যান্য Microservice সেই গুলো রিসিভ করে এবং প্রয়োজনে রিপ্লাই দেয়। এটা এক ধরনের Loosely Couple কারণ message গুলো কিউ তে থাকে।এতে message গুলোর কোন dependency থাকে না। তে বিভিন্ন ধরনের library ব্যবহার করা হয়।যেমনঃ RabbitMQ, Kafka।

Microservice এ সাধারণত অনেক গুলো ডিভাইসে এই সার্ভিস গুলো থাকে।এর ফলে তাদের maintain করা একটু জটিল হয়ে পড়ে। কারণ, একটি Microservice একটি server এ থাকে আর একটি Microservice অন্য একটি server এ থাকে । এতে তাদের সার্ভিস গুলো একই থাকে কিন্তু তাদের IP address গুলো পরিবর্তন হয়ে যায়। এর ফলে Microservice গুলোর মধ্যে যোগাযোগ করার জন্য এদের কে খুঁজে বের করতে হয়।এই খুঁজে পাওয়ার কাজ করে Discovery server এই server টা একেক সময় একেক IP তে থাকতে পারে, এর ফলে তার জন্য service registry মেইনটেইন করতে হয়। এটা মেইনটেইন করার জন্য কিছু library ব্যবহার করা হয়। যেমনঃ Eureka, Zookeeper ইত্যাদি।

একটি সিস্টেম বানানোর সময় যখন সিদ্ধান্ত নেয়া হয় তখন অনেকেই Microservice এর কথা চিন্তা করে না। কারন ছোট সিস্টেমের জন্য Microservice আসলেই কোন সমস্যা সমাধান করে না, কারন তখন এধরনের কোন সমস্যাই থাকে না। কিন্তু যখন সিস্টেমটি অনেক বড় হয়ে যায় এবং horizontal scaling এর প্রয়োজন পড়ে তখন নতুন করে Microservice ডেভেলপ করা অনেক খরচের ব্যাপার হয়ে দাড়ায় ।ভবিষ্যতের challenge আচ করতে পেরে অনেকেই শুরু থেকেই Microservice ডেভেলপ করতে উৎসাহিত করেন। যেমনঃNetflix, eBay এবং Amazon তাদের কয়েকটি মডিউল Microservice এ রূপান্তরিত করে ভবিষ্যতে এটির প্রয়োজন অনুধাবন করতে পারে এবং ধীরে ধীরে পুরো সিস্টেমটাকে Microservice এর একটি ইকোসিস্টেম ডেভেলপ করে।

এই ছিল আমাদের Microservice সম্পর্কে basic আলোচনা।বর্তমান এবং ভবিষ্যৎ প্রজন্মের প্রযুক্তির সাথে তাল মিলিয়ে চলার জন্য Microservice এর প্রয়োজনীয়তা তুলনাহীন । বর্তমান ডেভেলপমেন্ট চক্রের বিভিন্ন সমস্যা থেকেই আসলে এই ইকোসিস্টেমের আবির্ভাব।