Senkron ve Asenkron iletişim nedir örneklerle açıklayın?

Senkron iletişimde client bir istek gönderdiğinde, server bu isteği işler ve cevabı iletir. Client bu sürede bekler ve cevabı alır

Örneğin: HTTP protokolü kullanarak bir istek gönderildiğinde sunucu isteği alır, işler ve cevabı gönderir. Bu sürede client cevabı bekler.

Asenkron iletişimde ise client bir istek gönderdiğinde server bunu hemen işlemek yerine bir kuyruğa alır, bu kuyruğun sırasına göre (FIFO) işlenir. Bu sürede client işlemlerine devam edebilir.

Örneğin: Bir client e-posta gönderme isteğini server'a iletir. Server bunu kuyruğa alır ve zamanı geldiğinde epostayı gönderir. Client eposta'nın gönderilip gönderilmediğine bakmadan işlemlerine devam edebilir.

RabbitMQ ve Kafka arasındaki farkları araştırın?

RabbitMQ ve Kafka message brokers olarak nitelendirilir. Asenkron işlemlerde kullanılırlar.

RABBITMQ	KAFKA
Mesajları geçici veya kalıcı olarak saklayabilir	Mesajları diskte kalıcı olarak saklar
Genellikle tek bir consumer ve tek bir producer arasında gerçekleşir.	Genellikle bir kaç consumer'a yayınlanan topicl üzerinden mesajlar iletilir. Aynı mesaj birden fazla consumer tarafından tüketilebilir.
Küçük ölçekli kullanımda tercih edilir	Daha büyük ölçekli akışlar için tercih edilir

Docker ve VirtualMachine nedir?

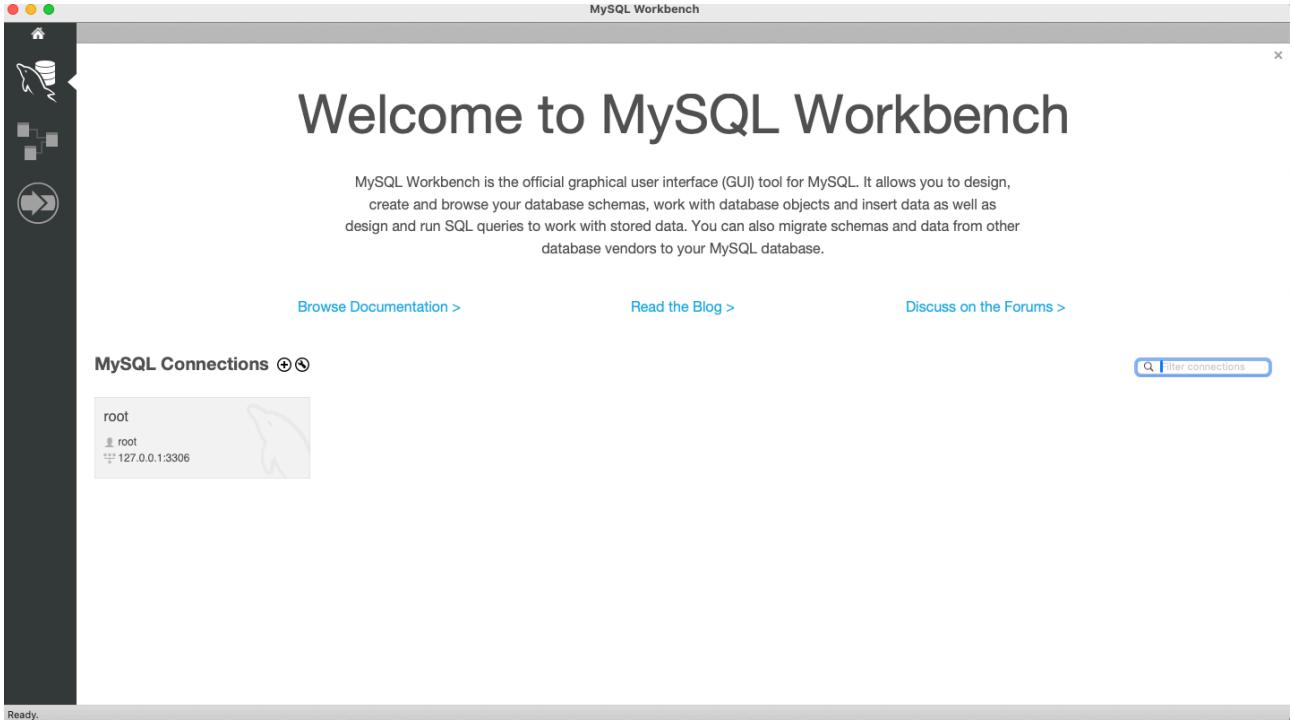
İkisi de yazılım uygulamalarını çalıştırmak için kullanılır.

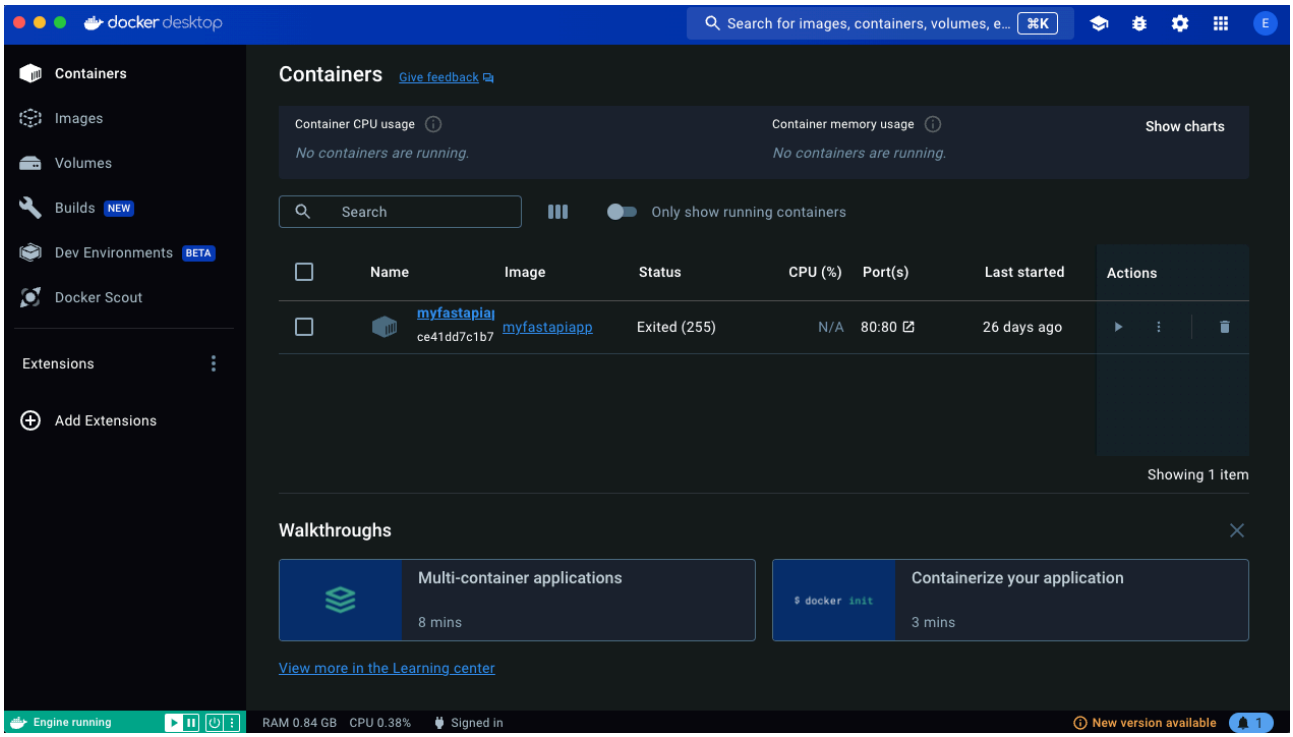
Docker:
Docker container => Uygulamanın çalışması için gerekli olan tüm dependency'ler içerir
Hızlı bir şekilde dağıtmak için ideal.

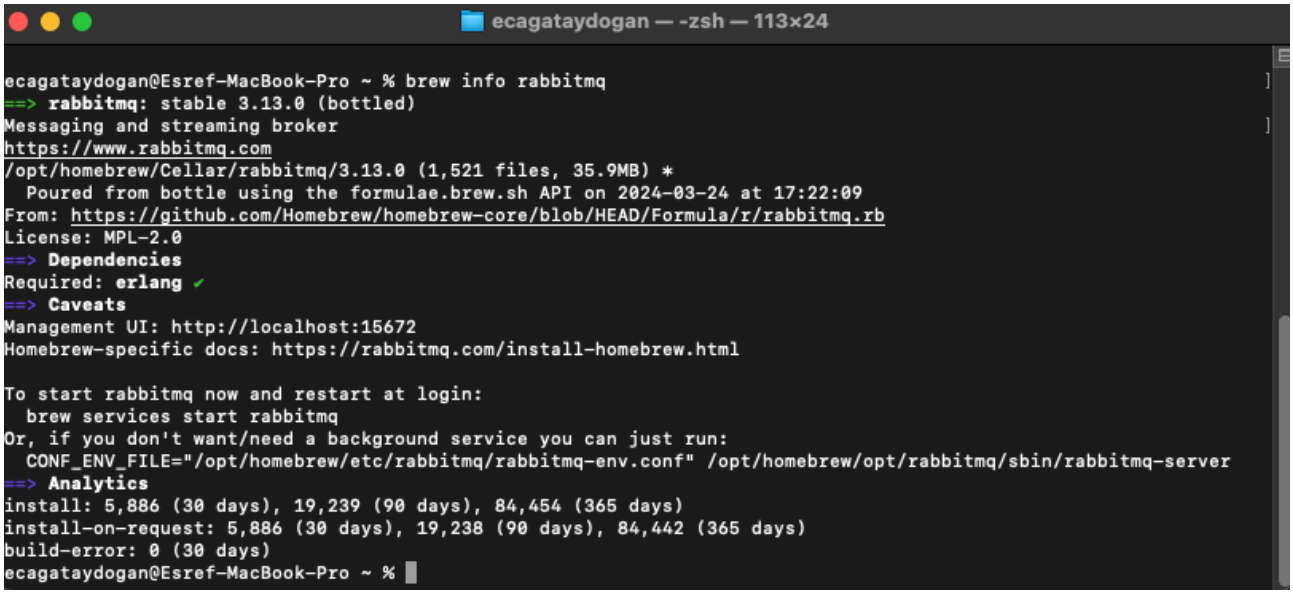
VirtualMachine:
Bir bilgisayar üzerinde sanal işletim sistemi çalıştırmak için kullanılan bir emülatördür.
Farklı işletim sistemlerinin aynı bilgisayarda, donanımda çalışması gereken durumlarda kullanılır.

Farklar:
Docker uygulama seviyesinde izolasyon sağlarken virtual machine'ler donanım seviyesinde izolasyon sağlarlar.
Docker daha hızlı başlatılabilir
Virtual machine daha fazla kaynak tüketir

Docker ile RabbitMQ ve PostgreSQL ve ya MySQL kurulumu yapın?







Docker komutlarını örneklerle açıklayın?

docker run => Container'ı çalıştırmak için
docker ps => Çalışan containerları listelemek için
docker stop => Çalışan container'ı durdurmak için
docker rm => Container'ı silmek için
docker build => Docker image'ı Dockerfile kullanarak oluşturmak için

Microservice ve Monotlith mimarilerini kıyaslayın?

Monolith:

Uygulamanın tüm bileşenleri tek bir kod tabanında olur. Dolayısıyla tek ve büyük bir uygulama olur.

Avantajlar:
Yönetmek daha kolay,
Farklı bileşenler arasında iletişim kurmak daha kolay,
Dağıtılacak tek bir servis var.

Dezavantajlar:
Bütün olduğu için ölçeklenebilirliği düşük
Tüm bileşenler aynı teknoloji ile yazılmalı.
Büyüdükçe karmaşıklaşır.

Microservice:

Uygulamanın farklı işlevselliklerini farklı servislere bölen bir yaklaşımdır.

Avantajlar:
Ölçeklenebilirliği yüksek çünkü her servis birbirinden bağımsız.
Farklı teknolojilerle geliştirilebilir. Tek bir teknolojiye bağımlı değil.

Dezavantajlar:
Daha karmaşık
Servirler arasında iletişim kurmak daha zor
Dağıtım için daha fazla konfigurasyon gerekli. Hepsi aynı ayrı deploy edilecek.

API Gateway, Service Discovery, Load Balancer kavramlarını açıklayın.

API Gateway:

Mikroservis mimarisinde giriş olarak kullanılır. Gelen istekleri alır ve yönlendirir. Genellikle security bu katmanda yapılır. Ayrıca loglama, trafiğin kontrol edilmesi gibi işlemler gerçekleştirilir.

Service Discovery:

Servislerin başlangıçta register olmasını gerektirir. Servislerin birbirini bulup iletişim kurması için gereklidir.
Client taleplerini yönlendirmek için kullanılır dynamic routing sağlar.

Load Balancer:

Ağdaki trafik yükünü dağıtan bir bileşendir. Gelen isteklerin hangi servislere iletileteceğini belirleyerek yükü dengeler.

Hibernate, JPA, Spring Data framework'lerini örneklerle açıklayın.

Hibernate:

Java tabanla uygulamalarda veritabanı işlemlerini gerçekleştirmek için kullanılan bir ORM framework'üdür. Dolayısıyla bir JPA implementasyonudur.

```
35 usages   esref
@Entity
@Table(name = "employee")
public class Employee {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    3 usages
    @Column(name ="first_name")
    private String firstName;

    3 usages
    @Column(name = "last_name")
    private String lastName;

    3 usages
    @Column(name = "email")
    private String email;
```

ORM: Bir java nesnesinin database'de bir tabloya karşılık gelmesi.

JPA:
ORM ile veritabanına erişim için kullanılan bir standart sağlar. Hibernate gibi frameworkler ise bunların standartlarını uygulanmış halidir.

Spring Data JPA:

Veritabanı işlemlerini kolaylaştırmak için kolaylık sağlar. Örneğin CRUD operasyonları için hazır arayüzler kullanılabilir