Soru 1: Docker nedir ve neden kullanılır? Linux işletim sistemi ile Docker arasındaki ilişki nedir?

- Yazılım uygulamalarının hızlı bir şekilde dağıtılmasını, yönetilmesini ve çalıştırılmasını sağlayan bir sanallaştırma platformudur. Docker, Linux çekirdeğinin özelliklerini kullanarak konteynerler oluşturur ve yönetir. Bu nedenle Docker'ın en iyi performansı ve uyumluluğu Linux üzerinde sağlanır.

Soru 2: Docker konteynerleri ile sanal makineler arasındaki temel farklar nelerdir?

- Docker konteynerleri sanal makinelerden daha hafif, daha hızlı ve kaynak tüketimi açısından daha verimlidir.

Soru 3: Bir Docker konteyneri nasıl oluşturulur ve çalıştırılır? Temel adımları açıklayın.

Docker'ın sistemde kurulu olduğunu varsayalım;

- "docker info" komutu ile durumu görüntüleriz.
- "docker run ngnix" komutu ile ngnix web sunucusunun son sürümünü indirip konteynerimizi oluşturuyoruz.
- "docker ps -a" komutu ile konteynerlerimizi listeliyoruz.

Soru 4: docker run komutunun işlevi nedir ve nasıl kullanılır? Örnek bir Docker konteyneri çalıştırma işlemi gerçekleştirin.

Bu komut, bir Docker imajından bir konteyner oluşturmaya yarar.

- "docker run ngnix" komutu ile ngnix web sunucusunun son sürümünü indirip konteynerimizi oluşturuyoruz.
- "docker ps -a" komutu ile konteynerlerimizi listeliyoruz.
- Konteyneri "exited" durumundan "up" durumuna getirmek için "container start konteyner-id" komutunu kullanıyoruz.
- "docker ps" komutunu kullaranak "up" durumunda ki konteynerlerimizi listeleyebiliriz.

Soru 5: docker ps komutu nedir ve çalışan Docker konteynerlerini nasıl listeler?

- Docker konteynerleri ile igili bilgileri listeler.Varsayılan olarak yalnızca çalışan (up) konteynerleri listeler.

Soru 6: Docker Hub nedir ve Docker konteynerleri nasıl paylaşılır veya indirilir?

- Docker konteynerlarını depolamak ve paylaşmak için kullanılan çevrimiçi bir kaynaktır.
- "docker pull imaj-ismi" komutu ile docker hubdan ihtiyacımız olan docker imajını indirebiliriz.
- "docker search aranacak-kelime" komutu ile de CLI ekranından imaj araması gerçekleştirebiliriz.

Soru 7: Bir Docker konteynerinin çalışan süreçlerini nasıl izlersiniz? docker exec komutunu kullanarak örnek bir işlemi nasıl kontrol edersiniz?

- "exec" çalışan bir docker konteynerine girmemizi ve bu konteyner içinde bir komut veya işlem çalıştırmanızı sağlar.
- "docker container exec –it konteyner-id bash" bu komut ile "up" durumundaki konteynerimize bash ile interaktif bir bağlantı sağlarız ve ihtiyacımız olan işlemleri gerçekleştirebiliriz.

Soru 8: Bir Docker konteynerinin içine nasıl giriş yapılır? docker exec veya docker attach gibi komutları kullanarak içeri girmenin farkları nelerdir?

- Docker konteynerinin içine girmek docker exec ve docker attach yöntemleri kullanılabilir.
- "docker exec" ile çalıştırılan işlem, konteynerin çalışmasına devam ederken, "docker attach" ile birlikte çalışan işlem, CTRL+C kullanarak sonlandırılabilir ve konteyner durur.

Soru 9: Docker komutları ile bir Docker imajının (image) nasıl oluşturulduğunu ve paylaşıldığını açıklayın.

- Docker imajı oluşturmak için bir dockerfile oluşturuyoruz. Dockerfile, docker imajının nasıl oluşturulacağını ve yapılandırılacağını tanımlıyoruz.
- "docker build -t my-custom-image" komutu ile dockerfile oluşturduktan sonra, bu dockerfile'ı kullanarak docker imajını oluşturuyoruz.
- Docker imajını docker hub veya diğer bir imaj deposuna yüklemek için "docker push" komutunu kullanıyoruz.

Soru 10: Bir Docker konteyneri çalışırken hangi ağ bağlantılarına sahiptir ve bu bağlantılar nasıl yönetilir?

- Docker konteynerleri aşağıdaki ağ bağlantılarına sahip olabilir ve bu bağlantılar docker tarafından otomatik olarak yönetilir:
  - Host Network
  - Bridge Network
  - Custom Bridge Network
  - Overlay Network
- "docker network create my-network" komutu ile ihtiyacımız olan network yapılandırmasını yönetebiliriz.

Soru 11: docker-compose nedir ve çoklu konteyner uygulamalarını nasıl yönetmek için kullanılır?

- Docker Compose, Docker kullanarak çoklu konteyner uygulamalarını daha kolay ve düzenli bir şekilde yönetmemize yardımcı olan bir şey. Bu araç uygulamamızı parçalara ayırmamıza ve bu parçaların nasıl çalışacağını belirlememize yardımcı olur.

Soru 12: Bir Docker konteynerinde çalışan bir uygulamanın bağımlılıklarını (dependencies) nasıl yönetirsiniz? Bir örnek verin.

- Diyelim ki, Python ile yazılmış bir web uygulaması çalıştırmak istiyorsunuz. Bu uygulama, Flask Framework kullanıyor ve bu nedenle Flask kütüphanesine bağımlıdır.
- "requirements.txt" adında bir dosya oluşturup Flask'ın kurulması için gerekli satırları bu txt nin içine yazarız. Dockerfile'ımızda "requirements.txt" referans vererek Flask bağımlılığını yönetmiş oluruz.

Soru 13: Docker konteynerlerinin güvenliği için neler yapabilirsiniz? Temel güvenlik önlemleri nelerdir?

- Docker imajlarımızın güvenlik yamalarını güncel tutarak bilinen güvenlik açıklarını kapatabiliriz.
- Konteynerlerimizi çalıştırmak için root hesabı yerine kullanıcı düzeyinde hesaplar oluşturabiliriz.
- İhtiyaç duymadığımız gereksiz uygulamaları kaldırıp, gereksiz servisleri ve portları kapatabiliriz.

Soru 14: Bir Docker imajını nasıl sıkıştırırsınız ve daha küçük bir boyuta nasıl getirirsiniz?

- Dockerfile içinde, sadece uygulamanın çalışması için gereken dosyaları kopyalayabilirz. İhtiyaç duyulmayan veya geçici dosyaları dahil etmeyiz.

Soru 15: Docker konteynerlerinin bir ağa nasıl bağlandığını ve bu konteynerlerin birbiriyle nasıl iletişim kurduğunu açıklayın.

- Docker konteynerlerinin ağa bağlanması veya birbirleriyle iletişim kurması için bir ağ oluşturabiliriz. "docker network create" komutunu kullanarak yeni bir ağ oluşturabilir veya varolan bir ağı kullanabiliriz.

Soru 16: Bir Docker imajının yapılandırma dosyalarını (örneğin, Dockerfile) nasıl oluşturursunuz ve düzenlersiniz?

```
FROM ubuntu:latest

COPY config /app/config

RUN apt-get update && apt-get install -y python3

COPY app.py /app/

CMD ["python3", "/app/app.py"]
```

- Örnek olarak bir Ubuntu imajından başlayarak bir Python uygulamasını çalıştıran bir Docker imajı oluşturur. İmaj, config dizinini ve app.py dosyasını imaja kopyalar, Python kurar ve uygulamayı çalıştırır.

Soru 17: Docker komutlarını ve işlemlerini otomatize etmek için neden Docker Compose veya Docker Swarm gibi araçlar kullanılır?

- Docker Compose ve Docker Swarm gibi araçlar, Docker'ın karmaşıklığını azaltır ve çoklu konteyner uygulamalarını daha kolay bir şekilde yönetmemizi sağlar.

Soru 18: Docker üzerinde çalışırken kaynak kullanımını (CPU, bellek vb.) izlemek ve yönetmek için hangi araçları kullanabilirsiniz?

Docker CLI, çalışan konteynerlerin kaynak kullanımını izlemek için "docker stats" komutu çalışan konteynerlerin CPU, bellek ve ağ kullanımını kullanabiliriz. "docker top" komutu ile konteynerin süreçlerini görüntüleyebiliriz.

Başka bir yöntem olarakda cAdvisor, Prometheus ve Grafana gibi araçlar kullanılabilir.

Soru 19: Docker ortamında bir web sunucusu (örneğin, Nginx) nasıl kurulur ve yapılandırılır?

- "docker pull nginx" komutunu kullanarak nginx Docker imajını indiririz.
- "docker run -d -p 80:80 --name my-nginx nginx" komutunu kullanarak nginx konteynerini başlatıp, local 80 numaralı portu konteynerin 80 numaralı portuna yönlendirip ve konteyneri my-nginx adıyla çalıştırmış oluruz.
- Varsayılan yapılandırmalarda değişiklik yapmak istersek "/etc/nginx/nginx.conf" dizininde ki konfigürasyon dosyasında vi editörü kullanrak gerekli değişiklikleri yapabiliriz.