

Containerized Applications and Clusters Questions & Answers

✓ What is the use of Container in Microservices?

Containers are a good way to manage microservice based application to develop and deploy them individually. You can encapsulate your microservice in a container image along with its dependencies, which then can be used to roll on-demand instances of microservice without any additional efforts required.

✓ What are containers?

Containers are used to provide consistent computing environment from a developer's laptop to a test environment, from a staging environment into production.

A container consists of an entire runtime environment: an application, plus all its dependencies, libraries and other binaries, and configuration files needed to run it, bundled into one package. Containerizing the application platform and its dependencies removes the differences in OS distributions and underlying infrastructure.

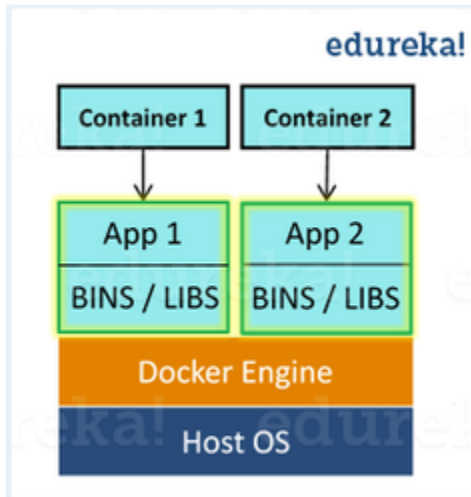
✓ What are the advantages that Containerization provides over virtualization?

- Containers provide real-time provisioning and scalability but VMs provide slow provisioning
- Containers are lightweight when compared to VMs
- VMs have limited performance when compared to containers
- Containers have better resource utilization compared to VMs

✓ What is Docker?

Docker is a containerization platform which packages your application and all its dependencies together in the form of containers so as to ensure that your application works seamlessly in any environment be it development or test or production.

Docker containers, wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries etc. anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment. **Unlike Virtual Machines which has its own OS Docker containers uses the host OS**



✓ What is Docker image?

Docker image is the source of Docker container. In other words, Docker images are used to create containers. Images are created with the build command, and they'll produce a container when started with run. Images are stored in a Docker registry such as registry.hub.docker.com because they can become quite large, images are designed to be composed of layers of other images, allowing a minimal amount of data to be sent when transferring images over the network.

✓ What is Docker container?

Docker containers include the application and all of its dependencies, but share the kernel with other containers, running as isolated processes in user space on the host operating system. Docker containers are not tied to any specific infrastructure: they run on any computer, on any infrastructure, and in any cloud. Docker containers can be created by either creating a Docker image and then running it or you can use Docker images that are present on the Dockerhub. Docker containers are basically runtime instances of Docker images.

✓ What is Docker hub?

Docker hub is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker cloud so you can deploy images to your hosts. It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline.

✓ What is Docker Swarm?

Docker Swarm is native clustering for Docker. It turns a pool of Docker hosts into a single, virtual Docker host. Docker Swarm serves the standard Docker API, any tool that already communicates with a Docker daemon can use Swarm to transparently scale to multiple hosts.

✓ What is Dockerfile used for?

Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build users can create an automated build that executes several command-line instructions in succession.

✓ How to create Docker container?

We can use Docker image to create Docker container by using the below command: `docker run -t -i command name`

✓ What is Docker Compose?

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration.

Using Compose is basically a three-step process:

1. Define your app's environment with a `Dockerfile` so it can be reproduced anywhere.
2. Define the services that make up your app in `docker-compose.yml` so they can be run together in an isolated environment.
3. Run `docker-compose up` and Compose starts and runs your entire app.

✓ What is Kubernetes?

Kubernetes is a container orchestration tool that is used for automating the managing, monitoring, scaling and deploying of containerized applications. It creates groups of containers that can be logical discovered and managed for easy operations on containers.

✓ What are the benefits of Kubernetes?

With container orchestration tool Kubernetes, it becomes extremely easy to handle the containers. You can respond to customer demands by deploying the applications faster and in a more predictable manner.

Here we will list some of the benefits of Kubernetes:

- Automatic scheduling
- Automated rollback
- Horizontal scaling
- Auto healing capabilities.

✓ What is the difference between Kubernetes and Docker Swarm?

Docker Swarm is the default container orchestration tool that comes with Docker. Docker Swarm can only orchestrate simple Docker containers. Kubernetes on the other hand helps to manage much more complex software application containers. Kubernetes offers support for larger demand production environment.

✓ What is orchestration in software?

Application orchestration in software process means you can integrate two or more applications. You will be able to automate arrangement, coordination and management of computer software. The goal of any orchestration process is to streamline and optimize frequent repeatable processes.

✓ What is a Pod in Kubernetes?

You can think of Kubernetes Pod as a group of containers that are run on the same host. So if you regularly deploy single containers then your container and Pod will be one and the same.

✓ What is a node in Kubernetes?

A node in Kubernetes is a worker machine which is also known as a minion. This node could be a physical machine or a virtual machine. For each of the node there is a service to run pods and it is managed by master components. The node services could include kubelet, kube-proxy and so on.

✓ What is a cluster in Kubernetes?

These master and node machines run the Kubernetes cluster orchestration system. A container cluster is the foundation of Container Engine: the Kubernetes objects that represent your containerized applications all run on top of a cluster. Overall, a container cluster lets you place and manage the containers in a dynamic setup.

✓ What is Minikube?

The Minikube makes it easy for the local running of Kubernetes. Within a Virtual Machine, the Minikube runs a single-node Kubernetes cluster.

✓ What is Kubectl?

The Kubectl is a Kubernetes command line tool that is used for deploying and managing applications on Kubernetes. The Kubectl is especially useful for inspecting the cluster resources, for creating, updating and deleting the components.

✓ What is a Cluster?

A group of two or more servers together ensuring availability of a service or application even when one of its members goes down.

✓ What's the advantage/disadvantage of having 1 node cluster?

One node cluster is used for situations wherein we just want the ability to get the stopped service restarted automatically. There are services which doesn't have the capability to restart on its own, they are hosted on one node cluster as the cluster service would restart the failed service and we are good to go. However if the node itself fails, the service becomes unavailable.

✓ Can 2 nodes belonging from multiple network subnet form a single Cluster?

While configuring a set of clustered nodes we need to have them on the same subnet.

✓ What kind of application is called cluster-aware?

An application is capable of being cluster-aware if it has the following characteristics:

- 1) It uses TCP/IP as a network protocol.
- 2) It maintains data in a configurable location.
- 3) It supports transaction processing.

✓ What is a kubernetes service?

A **Service in Kubernetes** is an abstraction which defines a logical set of Pods and a policy by which to access them. **Services** enable a loose coupling between dependent Pods. A **Service** is defined using YAML (preferred) or JSON, like all **Kubernetes** objects.

✓ What are various types of services in Kubernetes?

- **ClusterIP**: Exposes the service on a cluster-internal IP. Choosing this value makes the service only reachable from within the cluster. This is the default `ServiceType`.
- **NodePort**: Exposes the service on each Node's IP at a static port (the `NodePort`). A `ClusterIP` service, to which the `NodePort` service will route, is automatically created. You'll be able to contact the `NodePort` service, from outside the cluster, by requesting `<NodeIP>:<NodePort>`.
- **LoadBalancer**: Exposes the service externally using a cloud provider's load balancer. `NodePort` and `ClusterIP` services, to which the external load balancer will route, are automatically created.
- **ExternalName**: Maps the service to the contents of the `externalName` field (e.g. `foo.bar.example.com`), by returning a `CNAME` record with its value. No proxying of any kind is set up. This requires version 1.7 or higher of `kube-dns`

✓ What is Ingress in Kubernetes?

Kubernetes supports a high level abstraction called *Ingress*, which allows simple host or URL based HTTP routing. An ingress is a core concept (in beta) of Kubernetes, but is always implemented by a third party proxy. These implementations are known as ingress controllers.

An ingress controller typically doesn't eliminate the need for an external load balancer—the ingress controller simply adds an additional layer of routing and control behind the load balancer.

References:

<https://www.edureka.co/blog/interview-questions/microservices-interview-questions/>
<https://www.edureka.co/blog/interview-questions/docker-interview-questions/>
<https://docs.docker.com/compose/overview/>
<http://devopsonlinetraining.in/containerization-and-virtualization-interview-questions/>
<https://intellipaat.com/interview-question/kubernetes-interview-questions/>
<http://microsoftworld.blogspot.com/2011/07/questions-on-clustering.html>