**Docker: Rounak Bhadra**

Note: Question 1 to 9 each question is 1 marks, Question no. 10, 11 is 5 Marks while Question 12 is for 6 marks.

### Write a command to stopping and restarting a Docker container?

### Ans: For stopping Docker container : $ docker stop container\_id or $ docker stop container\_name

### For restarting Docker container : docker restart container\_id or $ docker restart container\_name

### Write a command to create a Docker image?

### 

### Ans : $ docker create --name tagname -p: portnumber base\_image\_name

### Example: docker create -v /data –name data ubuntu

### Or we can use $ docker build -t tagname docker file\_path

### Write a command to view all the running Docker containers?

### Ans: $ docker ps

### What command is used for running images as a container? Ans: To run it in foreground - $ docker run -it image\_name

### To run it in background - $ docker run -d image\_name

1. What is a cloud-hosted service of Docker providing registry capabilities for public and private content?

Ans: Docker Hub is the cloud-hosted service of Docker providing registry capabilities for public and private content .

1. What is a template used for describing a build of an image?

Ans: Dockerfile is the template used for describing a build of an image.

### Multiple containers running on a single machine all share the same resources such as the operating system kernel for instant boot and efficient utilization of RAM. True or False?

### Ans: True.

1. What command is used for remove all stopped containers, unused networks, build caches, and dangling images?  
   Ans: docker system prune  
   This command is used to remove all stopped containers, unused networks, build caches, and dangling images.
2. What command is used for running images as a container?

Ans: To run it in foreground - $ docker run -it image\_name

To run it in background - $ docker run -d image\_name

1. Explain what are Dockerfiles?

Ans: Dockerfile is the basic concept for building Docker images. It is better to understand it if you want to dig into Docker image building scenarios a bit more deeply.

Dockerfile is a text file that contains a list of commands (instructions), which describes how a Docker image is built based on them. The command “$docker build” tells Docker to build the image by following the content (instructions) inside the Dockerfile.

Dockerfile starts with a “From” command which indicates the base image. The subsequent commands in the Docker file are executed on the base image which must be a valid image.

First of all we need to create the dockerfile itself, then we need to run the docker build command with the dockerfile to create the docker image.

To build your Docker image, you could call “$docker image build”command by specifying building directory with the Dockerfile in the terminal and some arguments can be provided as well.

For example: to install multiple packages using single dockerfile (git and firefox)

FROM centos

MAINTAINER [rounakbhadra@gmail.com](mailto:rounakbhadra@gmail.com)

RUN yum install -y \

git \ firefox

### List the most commonly used instructions in Dockerfile?

### Ans: We all know the importance of dockerfile in creating an efficient and flexible Docker Image. A dockerfile contains a set of instructions that are executed step by step when you use the docker build command to build the docker image. It contains certain instructions and commands that decides the structure of your image, the amount of time taken to build the image, contains instructions related to docker build context, contains information related to the packages and libraries to be installed in the container and many more. Hence, it becomes very important to create an efficient, reusable, clean dockerfile as it contains the blueprint of the image that you will build.

* FROM: Initializes a new build stage and sets the Base Image

$ FROM <image name>:<tag name>

* RUN: Will execute any commands in a new layer  
  $RUN apt −y install chrome

$RUN apt −y update

* CMD: Provides a default for an executing container.

$ CMD echo "Rounak dockerfile"

* LABEL: Adds metadata to an image

$ LABEL description="This is a example"

* EXPOSE: Informs Docker that the container listens on the specified network ports at runtime

$ Expose 8020

* ENV: Sets the environment variable<key>to the value<value>
* ADD: Copies new files, directories or remote file URLs from<src> and adds them to the filesystem of the image at the path<dest>.

$ ADD /Desktop/myapp/rounak.txt /usr/src/app

* COPY: Copies new files or directories from<src>and adds them to the filesystem of the container at the path<dest>.

FROM ubuntu

WORKDIR /rivu/src/app

COPY ∽/Desktop/user

* ENTRYPOINT: Allows for configuring a container that will run as an executable
* VOLUME: Creates a mount point with the specified name and marks it as holding externally mounted volumes from native host or other containers

### Explain Docker lifecycle of Docker Container?

### Ans: A Docker image is a static model containing a preinstalled application. When running this image, container is created. An image can be used to start as many containers as needed. The following diagram shows the life cycle of a Docker container and associated commands.

There are two ways of getting a Docker image:

* Retrieve a Docker image from a registry (often Docker Hub).
* Build it from a Dockerfile. All standard Jahia and jCustomer Docker images are available as public Docker Hub images.

To list all running containers, execute this Docker command:

docker ps

To list all the containers on your machine, including the ones that are stopped or exited:

docker ps -a

Note: For more options, refer to the

To run a container in detached mode:

docker run -p 8080:8080 -d

The command executes the container and returns the container’s UUID. You may need the command if you have not named your container.

In detached mode, you can still display the standard output logs of the container in a console by executing this Docker command:

docker logs -f {container\_id}

To stop a container:

docker stop {container\_id}

To start a container:docker start {container\_id}

To delete a container:docker rm {container\_id}

To remove and start a new container with a given name: docker run -p 8080:8080 --rm--name {container name}

### 

### Kubenetes:

### Each Question is 1 marks.

1. **At its core, Kubernetes is a platform for:**

### Provisioning machines (similar to Puppet, Ansiblee

### Running and scheduling container applications on a cluster.

### Packaging software in containers

### **Ans: B**

**2) Which of the following are core Kubernetes objects ?**

A. Pods

B.Volumes

C.Services

D.All of them

**Ans: D**

**3)Kubernetes Network proxy runs on which node ?**

A. Master Node

B.Worker Node

C.CIDR Node

D.Both A & B

**Ans: D**

**4) runs on each node and ensures containers are running in a pod.**

A. Etcd

B.Pod

C.Kubelet

D.Scheduler

**Ans: C**

**5) Which of them is a Kubernetes Controller ?**

A. ReplicaSet

B.Deployment

C.Rolling Updates

D.Both A & B

**Ans: D**

**6)....................... are the Kubernetes controllers.**

A. Replicaset

B.Deployment

C.Namespace

D.Both Replicaset & Deployment

**Ans: D**

**7) Kubernetes is the type of cluster management software.**

A. True

B.False

**Ans: A**

**8) Which of the following are the components of Kubernetes Master Machine?**

A. Scheduler

B.Controller Manager

C.API Server & etcd

D.All of the above

**Ans: D**

**9) Kubernetes API currently supports ........... type of selectors.**

A. Set-based selectors

B.Equality-based selectors

C.both Set-based & Equality-based selectors

D.None of the above

**Ans: C**

**10) What are the some important functionalities of a Namespace in Kubernetes?**

A. Namespaces help pod-to-pod communication using the same namespace.

B.Namespaces provide logical separation between the teams and their environments.

C.Namespaces are virtual clusters that can sit on top of the same physical cluster.

D.All of the above

**Ans: A**

**11) There are ............ types of Pods in kubernets?**

A. 2

B.3

C.4

D.6

**Ans: D**

**12) GKE stands for \_\_\_\_.**

A. Google Cluster Engine

B. Google Kubernetes Engine

C. Google Container Engine

D. None of the above

**Ans: B**

**13) Which of the following commands allow you to validate a cluster created with Kubernetes operations?**

A. kubectl validate cluster

B. kubeadm validate cluster

C. kops validate cluster

D. None of the above

**Ans: B**

**14) What is the default range of ports used to expose a NodePort service?**

A. 30000-32767

B. 500-1000

C. 60000-65536

D. 1024-32767

**Ans: A**

**15) Which of the following commands gives you detailed info on a Pod?**

A. kubectl describe pods

B. kubectl get pods -vvv

C. kubectl get pods –detail

D. kubectl pods inspect

**Ans: A**

**16) What is the default protocol for a Service?**

A. TCP

B. UDP

C. HTTP

D. SSH

**Ans: A**

**17) Which of the following is true about Pods and IP addressing?**

A. Pods only work with IPv6 addresses

B. All containers in a Pod get unique IP addresses

C. An external DHCP server is required for Pod IP addressing

D. All containers in a Pod share a single IP address

**Ans: D**

**18) In Kubernetes, a node is:**

A. A tool for starting a kubernetes cluster on a local machine  
B. A worker machine  
C. A machine that coordinates the scheduling and management of application containers on the cluster

**Ans: B**

**19) What can you deploy on Kubernetes?**

A. Containers  
B. Virtual Machines  
C. System Processes (like sshd, httpd)

**Ans: A**

20) **We have a node named nodeA, and we want to add a tainting effect to it, which command we will use**

a. kubectl taint nodes nodeA key:=NoSchedule

b. kubectl taint node key:=NoSchedule

c. kubectl taint nodes nodeA

d. All of the Above

**Ans: A**