

[Dashboard](#) / [My courses](#) / [Computer Engineering](#) / [CEIT-Odd-sem-22](#) / [Devops-odd-sem-22](#) / [10 October - 16 Octo...](#) / [Midsem Ex...](#)

**Started on** Friday, 14 October 2022, 12:03 PM

**State** Finished

**Completed on** Friday, 14 October 2022, 1:48 PM

**Time taken** 1 hour 45 mins

**Grade** 35.05 out of 40.00 (87.62%)

Question 1

Complete

Mark 0.50 out of 1.00

Select correct statements about the command kubectl

- a. is used to manage pods on a single node cluster on your laptop
- b. in minikube installation the kubectl is already pre-configured to talk to kube-apiserver
- c. talks to kube-apiserver
- d. runs on the kubernetes user's laptop

The correct answers are: runs on the kubernetes user's laptop, talks to kube-apiserver, in minikube installation the kubectl is already pre-configured to talk to kube-apiserver, is used to manage pods on a single node cluster on your laptop

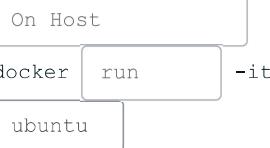
**Question 2**

Complete

Mark 1.82 out of 2.00

Suppose it is required to create a custom ubuntu docker image, saved as a tar file, with the base ubuntu and the package "vim" into it.

Complete the following description of commands, which aim to achieve the above aim.



On container prompt

```
apt
update
```

```
apt
install
```

vim

On Host, while container is running

```
docker
images
```

# get list of images. Output shown below.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<none>	<none>	55fb3b7c6802	5 seconds ago	116MB
my-fedora	latest	1862a1e17ccb	3 days ago	434MB
capitalserver	latest	e4c2e09a1ef7	3 weeks ago	124MB

```
docker
commit
55fb3b7c6802

ubuntu-vim
docker save
ubuntu-vim
-o /tmp/ubuntu-vim.tar
```

**Question 3**

Complete

Mark 1.00 out of 1.00

## Math pairs

node affinity A property of Pods, attracting it to specified nodes

Taint Nodes repelling a set of pods

Toleration Let the scheduler schedule a pod with matching taints

Your answer is correct.

The correct answer is: node affinity → A property of Pods, attracting it to specified nodes, Taint → Nodes repelling a set of pods, Toleration → Let the scheduler schedule a pod with matching taints

**Question 4**

Complete

Mark 1.60 out of 2.00

Which of the following images (for x86a and linux) are downloaded less than a billion times from docker-hub?

- a. haproxy
- b. rabbitmq
- c. golang
- d. httpd
- e. elasticsearch
- f. php
- g. busybox
- h. tomcat
- i. redis
- j. nodejs

The correct answers are: rabbitmq, elasticsearch, haproxy, php, tomcat

**Question 5**

Complete

Mark 1.33 out of 2.00

Consider the following sequence of commands

```
git init  
vi a.c # file edited /* A */  
git add a.c /* B */  
vi a.c # file edited /* C */  
git add a.c  
git commit /* D */  
vi a.c /* E */  
git add a.c /* F */  
git commit
```

For each of the lines, labeled as A to F, select the proper option describing the state of the file a.c

- A untracked
- B staged
- C modified
- F unmodified
- E unmodified
- D unmodified

Your answer is partially correct.

You have correctly selected 4.

The correct answer is: A → untracked, B → staged, C → modified, F → staged, E → modified, D → unmodified

**Question 6**

Complete

Mark 2.00 out of 2.00

Given below is the output of one command:

```
$ kubectl describe replicaset.apps nginx-deployment
Name:           nginx-deployment-6768c68f7b
Namespace:      default
Selector:       app=nginx,pod-template-hash=6768c68f7b
Labels:         app=nginx
                pod-template-hash=6768c68f7b
Annotations:    deployment.kubernetes.io/desired-replicas: 3
                deployment.kubernetes.io/max-replicas: 4
                deployment.kubernetes.io/revision: 2
Controlled By: Deployment/nginx-deployment
Replicas:       3 current / 3 desired
Pods Status:    3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=nginx
          pod-template-hash=6768c68f7b
Containers:
  nginx:
    Image:      nginx:1.20
    Port:       <none>
    Host Port:  <none>
    Environment: <none>
    Mounts:     <none>
    Volumes:    <none>
Events:
  Type      Reason        Age      From            Message
  ----      ----        ---      ----            -----
  Normal    SuccessfulCreate 3m7s    replicaset-controller  Created pod: nginx-deployment-6768c68f7b-sqgbf
  Normal    SuccessfulCreate 2m43s   replicaset-controller  Created pod: nginx-deployment-6768c68f7b-xbtr8
  Normal    SuccessfulCreate 2m41s   replicaset-controller  Created pod: nginx-deployment-6768c68f7b-pq2nt
Name:           nginx-deployment-77979d4865
Namespace:      default
Selector:       app=nginx,pod-template-hash=77979d4865
Labels:         app=nginx
                pod-template-hash=77979d4865
Annotations:    deployment.kubernetes.io/desired-replicas: 3
                deployment.kubernetes.io/max-replicas: 4
                deployment.kubernetes.io/revision: 1
Controlled By: Deployment/nginx-deployment
Replicas:       0 current / 0 desired
Pods Status:    0 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=nginx
          pod-template-hash=77979d4865
Containers:
  nginx:
    Image:      nginx:1.18
    Port:       <none>
    Host Port:  <none>
    Environment: <none>
```

```

Mounts: <none>
Volumes: <none>
Events:
  Type    Reason          Age     From            Message
  ----  -----  ----  -----
Normal  SuccessfulCreate  3m51s  replicaset-controller  Created pod: nginx-deployment-77979d4865-
jskrv
Normal  SuccessfulCreate  3m51s  replicaset-controller  Created pod: nginx-deployment-77979d4865-
bmwj7
Normal  SuccessfulCreate  3m51s  replicaset-controller  Created pod: nginx-deployment-77979d4865-
tfjrs
Normal  SuccessfulDelete  2m43s  replicaset-controller  Deleted pod: nginx-deployment-77979d4865-
jskrv
Normal  SuccessfulDelete  2m41s  replicaset-controller  Deleted pod: nginx-deployment-77979d4865-
bmwj7
Normal  SuccessfulDelete  2m39s  replicaset-controller  Deleted pod: nginx-deployment-77979d4865-
tfjrs

```

Mark those statements as True, which can be deduced from the information given in the above output.

True	False	
<input checked="" type="radio"/>	<input type="radio"/>	Number of replicas is 3
<input type="radio"/>	<input checked="" type="radio"/>	There are 3 replica-sets
<input checked="" type="radio"/>	<input type="radio"/>	nginx was upgraded from 1.18 to 1.20
<input checked="" type="radio"/>	<input type="radio"/>	Pod has only one container, that is nginx in it
<input checked="" type="radio"/>	<input type="radio"/>	The latest version of nginx is running right now
<input type="radio"/>	<input checked="" type="radio"/>	nginx was downgraded from 1.20 to 1.18
<input type="radio"/>	<input checked="" type="radio"/>	The upgrade/downgrade process is in operation and not complete yet

Number of replicas is 3: True

There are 3 replica-sets: False

nginx was upgraded from 1.18 to 1.20: True

Pod has only one container, that is nginx in it: True

The latest version of nginx is running right now: True

nginx was downgraded from 1.20 to 1.18: False

The upgrade/downgrade process is in operation and not complete yet: False

**Question 7**

Complete

Mark 0.75 out of 2.00

Select all the incorrect statements about git merge and git rebase.

Select one or more:

- a. git merge may not work automatically and fail
- b. git merge necessarily creates a commit
- c. git rebase creates a more linear history
- d. we can't use git unless we resolve a failed rebase
- e. git rebase may not work automatically and fail
- f. we can't use git unless we resolve a failed merge
- g. git merge creates a more non-linear history
- h. merge and rebase are just two names for the same thing
- i. rebase is for newbees, and merge for experienced programmers.
- j. rebase can lead to very complicated scenarios in distributed development.
- k. merge is costlier than a rebase
- l. rebase is costlier than a merge

Your answer is partially correct.

You have selected too many options.

The correct answers are: merge and rebase are just two names for the same thing, merge is costlier than a rebase, rebase is costlier than a merge, rebase is for newbees, and merge for experienced programmers.

**Question 8**

Complete

Mark 2.00 out of 2.00

Match the actions with each Kubernetes component, for the purpose of creation of a pod

detect that new pod object is created

kube-scheduler

authentication request is validated

kube-apiserver

authentication request is generated

kubectl

register a node with cluster

kubelet

application image is deployed

container-runtime

send reports of pods regularly to kube-apiserver

kubelet

node to run the pod is identified

kube-scheduler

pod is created on the node

kubelet

pod object is created

kube-apiserver

request docker to run the instance

kubelet

pod info is updated in etcd

kube-apiserver

The correct answer is: detect that new pod object is created → kube-scheduler, authentication request is validated → kube-apiserver, authentication request is generated → kubectl, register a node with cluster → kubelet, application image is deployed → container-runtime, send reports of pods regularly to kube-apiserver → kubelet, node to run the pod is identified → kube-scheduler, pod is created on the node → kubelet, pod object is created → kube-apiserver, request docker to run the instance → kubelet, pod info is updated in etcd → kube-apiserver

**Question 9**

Complete

Mark 0.80 out of 1.00

Select the correct statements, which describe why the ClusterIP is used, how it is used and what it does.

- a. ClusterIP service is reachable only within the cluster
- b. Using Pod's IP address leads to non-portability, as when Pod is deleted and respawned, it will have a different IP address
- c. If ClusterIP goes down, then it's respawned but with same IP address, so the existing connections are maintained
- d. ClusterIP service in Kubernetes is a REST object, similar to a Pod
- e. The ClusterIP exports the IP address to which the clients of the service attach

Your answer is partially correct.

You have correctly selected 4.

The correct answers are: Using Pod's IP address leads to non-portability, as when Pod is deleted and respawned, it will have a different IP address, ClusterIP service in Kubernetes is a REST object, similar to a Pod, The ClusterIP exports the IP address to which the clients of the service attach, If ClusterIP goes down, then it's respawned but with same IP address, so the existing connections are maintained, ClusterIP service is reachable only within the cluster

**Question 10**

Complete

Mark 0.83 out of 1.00

Consider following the sequence of commands executed either on the host or container or somewhere so that all of them together make some sense.

```
docker run -it ubuntu
apt update; apt install net-tools iputils-ping openssh-server openssh-client
docker commit <container-id> myubuntu
exit

docker network create network1
docker run -it -v /tmp/folder:/folder --network network1 --hostname u1 myubuntu
echo 1 >> /folder/1
docker run -it -v /tmp/folder:/folder --network network1 --hostname u2 myubuntu
echo 2 >> /folder/1
docker run -it -v /tmp/folder:/folder --network network1 --hostname u3 myubuntu
echo 3 >> /folder/1
```

Mark statements as True/False w.r.t. above commands.

---

<b>True</b>	<b>False</b>
-------------	--------------

---

- |                                  |                                  |   |
|----------------------------------|----------------------------------|---|
| <input checked="" type="radio"/> | <input type="radio"/>            | The file /tmp/folder/1 on host finally contains the data "1\n2\n3\n"                            |
| <input checked="" type="radio"/> | <input type="radio"/>            | u1 u2 u3 will be able to ping each other  |
| <input checked="" type="radio"/> | <input type="radio"/>            | The file /folder/1 on container u1 finally contains the data "1\n2\n3\n"                        |
| <input checked="" type="radio"/> | <input type="radio"/>            | The creation of a network does not serve any purpose in the echo commands                       |
| <input type="radio"/>            | <input checked="" type="radio"/> | The openssh-server installed in the image has helped the containers communicate with each other |
| <input checked="" type="radio"/> | <input type="radio"/>            | ssh from u1 to u2 will work now.  |
- 

The file /tmp/folder/1 on host finally contains the data "1\n2\n3\n": True

u1 u2 u3 will be able to ping each other: True

The file /folder/1 on container u1 finally contains the data "1\n2\n3\n": True

The creation of a network does not serve any purpose in the echo commands: True

The openssh-server installed in the image has helped the containers communicate with each other: False

ssh from u1 to u2 will work now.: False

**Question 11**

Complete

Mark 1.50 out of 1.50

Consider the following declaration in a YAML file

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: myapp-ha
  labels:
    app: myapp
    type: front-end
spec:
  template:
    metadata:
      name: myapp
      labels:
        app: myapp
        type: frontend
    spec:
      containers:
        - name: container-1
          image: redis
        - name: container-2
          image: httpd
  replicas: 3
  selector:
    matchLabels:
      type: frontend
```

This configuration will

create

6

containers

create

3

pods

create a replicaset

**Question 12**

Complete

Mark 2.00 out of 2.00

Select all the correct statements about branches in git

Select one or more:

- a. master branch is treated specially by git
- b. creating a branch involves copy of the commits
- c. one can work in parallel on different branches
- d. 'master' is just another branch
- e. switching branches is a costly operation
- f. creation of a branch results in a commit
- g. creating a branch is cheap with git
- h. a branch is just a pointer in git data-structure

Your answer is correct.

The correct answers are: a branch is just a pointer in git data-structure, creating a branch is cheap with git, 'master' is just another branch, one can work in parallel on different branches

**Question 13**

Complete

Mark 2.00 out of 2.00

Write a Dockerfile that does the following

- (a) uses a Ubuntu base image
- (b) Add the binary "hello" available in current directory to the /usr/local/bin/ path in the image
- (c) Installs the tree command also
- (d) Installs apache in the image
- (e) Runs apache on port 3030

```
FROM ubuntu
COPY ./hello /usr/local/bin/
RUN apt update
RUN apt install tree
RUN apt install apache2
EXPOSE 3030
```

Comment:

**Question 14**

Complete

Mark 1.50 out of 1.50

Which of the following statements are true about Kube API Server?

- a. kube-apiserver deploys a pod on a node
- b. pod objects are created by kube-apiserver, but without assigning to a node
- c. kube-apiserver updates the data in 'etcd' for all pods
- d. kube-apiserver schedules a pod on a node
- e. Non kubernetes applications can also connect the kube-apiserver using HTTP protocol
- f. Each component of kubernetes interacts with the kube-apiserver

The correct answers are: Each component of kubernetes interacts with the kube-apiserver, Non kubernetes applications can also connect the kube-apiserver using HTTP protocol, pod objects are created by kube-apiserver, but without assigning to a node, kube-apiserver updates the data in 'etcd' for all pods

**Question 15**

Complete

Mark 1.71 out of 2.00

JSON files were extracted from the tar image of a docker.

The files are listed below:

**File1**

```
{  
    "architecture": "amd64",  
    "config": {  
        "Hostname": "896ed4258c84",  
        "Domainname": "",  
        "User": "",  
        "AttachStdin": true,  
        "AttachStdout": true,  
        "AttachStderr": true,  
        "Tty": true,  
        "OpenStdin": true,  
        "StdinOnce": true,  
        "Env": [  
            "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"  
        ],  
        "Cmd": [  
            "bash"  
        ],  
        "Image": "ubuntu",  
        "Volumes": null,  
        "WorkingDir": "",  
        "Entrypoint": null,  
        "OnBuild": null,  
        "Labels": {}  
    },  
    "container": "896ed4258c8422489a183fc1a0f7d110e4e5c432fc5d62971b24015d58620701",  
    "container_config": {  
        "Hostname": "896ed4258c84",  
        "Domainname": "",  
        "User": "",  
        "AttachStdin": true,  
        "AttachStdout": true,  
        "AttachStderr": true,  
        "Tty": true,  
        "OpenStdin": true,  
        "StdinOnce": true,  
        "Env": [  
            "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"  
        ],  
        "Cmd": [  
            "bash"  
        ],  
        "Image": "ubuntu",  
        "Volumes": null,  
        "WorkingDir": "",  
        "Entrypoint": null,  
        "OnBuild": null,  
        "Labels": {}  
    },  
    "created": "2022-09-21T15:31:32.742756984Z",  
    "docker_version": "20.10.18",  
    "history": [  
        {  
            "created": "2022-09-01T23:46:35.026691064Z",  
            "created_by": "/bin/sh -c #(nop) ADD file:a7268f82a86219801950401c224cabbdd83ef510a7c71396b25f70c2639ae4fa in / "  
    ]  
}
```

```

},
{
  "created": "2022-09-01T23:46:35.375057619Z",
  "created_by": "/bin/sh -c #(nop) CMD [\"bash\"]",
  "empty_layer": true
},
{
  "created": "2022-09-21T15:31:32.742756984Z",
  "created_by": "bash"
}
],
"os": "linux",
"rootfs": {
  "type": "layers",
  "diff_ids": [
    "sha256:7f5cbd8cc787c8d628630756bcc7240e6c96b876c2882e6fc980a8b60cdaf274",
    "sha256:0a7af9a8e086cf665e913c752013f6c918fc132485e3b152b538179f0314467"
  ]
}
}
}

```

**file2:**

```
{
  "id": "03c159c5eed07601110c82ad724cfe7b85eb6c9c9a91e44775338053e8e988a",
  "created": "1970-01-01T05:30:00+05:30",
  "container_config": {
    "Hostname": "",
    "Domainname": "",
    "User": "",
    "AttachStdin": false,
    "AttachStdout": false,
    "AttachStderr": false,
    "Tty": false,
    "OpenStdin": false,
    "StdinOnce": false,
    "Env": null,
    "Cmd": null,
    "Image": "",
    "Volumes": null,
    "WorkingDir": "",
    "Entrypoint": null,
    "OnBuild": null,
    "Labels": null
  },
  "os": "linux"
}
```

**File3:**

```
{
  "id": "e81f28d0db59ae988d99e9484ef94debd8b84cc4afe9159a4816aeb0febdd1cd",
  "parent": "03c159c5eed07601110c82ad724cfe7b85eb6c9c9a91e44775338053e8e988a",
  "created": "2022-09-21T15:31:32.742756984Z",
  "container": "896ed4258c8422489a183fc1a0f7d110e4e5c432fc5d62971b24015d58620701",
  "container_config": {
    "Hostname": "896ed4258c84",
    "Domainname": "",
    "User": "",
    "AttachStdin": true,
    "AttachStdout": true,
    "AttachStderr": true,
    "Tty": true,
    "OpenStdin": true,
    "StdinOnce": true,

```

```

"Env": [
    "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
],
"Cmd": [
    "bash"
],
"Image": "ubuntu",
"Volumes": null,
"WorkingDir": "",
"Entrypoint": null,
"OnBuild": null,
"Labels": {}
},
"docker_version": "20.10.18",
"config": {
    "Hostname": "896ed4258c84",
    "Domainname": "",
    "User": "",
    "AttachStdin": true,
    "AttachStdout": true,
    "AttachStderr": true,
    "Tty": true,
    "OpenStdin": true,
    "StdinOnce": true,
    "Env": [
        "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
    ],
    "Cmd": [
        "bash"
    ],
    "Image": "ubuntu",
    "Volumes": null,
    "WorkingDir": "",
    "Entrypoint": null,
    "OnBuild": null,
    "Labels": {}
},
"architecture": "amd64",
"os": "linux"
}

```

**file4:**

```

[
{
    "Config": "aef552620a0f9b00184fd236648d3ec745e50126a6ac82a0117b3ed517296280.json",
    "RepoTags": [
        "my-ubuntu-tar:latest"
    ],
    "Layers": [
        "03c159c5eedad07601110c82ad724cf7b85eb6c9c9a91e44775338053e8e988a/layer.tar",
        "e81f28d0db59ae988d99e9484ef94debd8b84cc4afe9159a4816aeb0febdd1cd/layer.tar"
    ]
}
]

```

Mark those statements as True which can be definitely deduced from this information. Wrong/incomplete deductions should be marked as False.

---

<b>True</b>	<b>False</b>
-------------	--------------

---




The base image is fedora

---

**True      False**

The "id" of base image is  
896ed4258c8422489a183fc1a0f7d110e4e5c432fc5d  
62971b24015d58620701

There is at least one layer added on top of base  
image

The "id" of base image is  
03c159c5eed07601110c82ad724cf7b85eb6c9c9a9  
1e44775338053e8e988a

The image contains the command "tar" in it.

The base image is ubuntu

The "id" of base image is  
aef552620a0f9b00184fd236648d3ec745e50126a6ac  
82a0117b3ed517296280

The base image is fedora: False

The "id" of base image is 896ed4258c8422489a183fc1a0f7d110e4e5c432fc5d62971b24015d58620701: False

There is at least one layer added on top of base image: True

The "id" of base image is 03c159c5eed07601110c82ad724cf7b85eb6c9c9a91e44775338053e8e988a: True

The image contains the command "tar" in it.: False

The base image is ubuntu: True

The "id" of base image is aef552620a0f9b00184fd236648d3ec745e50126a6ac82a0117b3ed517296280: False

**Question 16**

Complete

Mark 2.00 out of 2.00

Consider the following command and its output:

```
$ kubectl describe replicaset myapp-ha
Name:           myapp-ha
Namespace:      default
Selector:       type=frontend
Labels:         app=myapp
                type=front-end
Annotations:    <none>
Replicas:       3 current / 3 desired
Pods Status:   2 Running / 1 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=myapp
            type=frontend
Containers:
  container-1:
    Image:      redis
    Port:       <none>
    Host Port: <none>
    Environment: <none>
    Mounts:     <none>
  container-2:
    Image:      httpd
    Port:       <none>
    Host Port: <none>
    Environment: <none>
    Mounts:     <none>
    Volumes:    <none>
Events:
  Type      Reason        Age      From            Message
  ----      -----        ---      ----            -----
  Normal    SuccessfulCreate 24s     replicaset-controller  Created pod: myapp-ha-7kkzl
  Normal    SuccessfulCreate 24s     replicaset-controller  Created pod: myapp-ha-w7pnp
  Normal    SuccessfulCreate 24s     replicaset-controller  Created pod: myapp-ha-rsj4n
```

Followed by few commands, and in the end this output:

```
$ kubectl describe replicaset myapp-ha
Name:           myapp-ha
Namespace:      default
Selector:       type=frontend
Labels:         app=myapp
                type=front-end
Annotations:    <none>
Replicas:       3 current / 3 desired
Pods Status:   0 Running / 3 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=myapp
            type=frontend
Containers:
  container-1:
    Image:      redis
    Port:       <none>
    Host Port: <none>
    Environment: <none>
```

```
Mounts:          <none>
container-2:
  Image:        httpd
  Port:         <none>
  Host Port:   <none>
  Environment: <none>
  Mounts:       <none>
  Volumes:     <none>

Events:
  Type  Reason          Age   From           Message
  ----  -----          ---  --            -----
Normal  SuccessfulCreate 93s  replicaset-controller  Created pod: myapp-ha-7kkz1
Normal  SuccessfulCreate 93s  replicaset-controller  Created pod: myapp-ha-w7pnp
Normal  SuccessfulCreate 93s  replicaset-controller  Created pod: myapp-ha-rsj4n
Normal  SuccessfulCreate 25s  replicaset-controller  Created pod: myapp-ha-dwg7r
Normal  SuccessfulCreate  5s   replicaset-controller  Created pod: myapp-ha-dt7nd
Normal  SuccessfulCreate  5s   replicaset-controller  Created pod: myapp-ha-plw49
Normal  SuccessfulCreate  5s   replicaset-controller  Created pod: myapp-ha-tzb9p
```

Here,

the number of pods defined in the replicaset are:

3

the number of pods deleted so far are :

4

the first re-created pod had the ID:

myapp-ha-dwg7r

At the time of the last command and its output, the number of pods waiting is 3. That means:

3 deleted pods are being re-created

**Question 17**

Complete

Mark 0.90 out of 1.00

Select all the statements that correctly identify the need, use, and limitations for containers.

- a. Changes in a shared library could break applications, but containers simplify upgrade and rollback.
- b. Containers offer more elasticity, reusability
- c. Isolated sandbox environment helps in more reliability and uptime
- d. Managing library version dependency is a major concern, and containers ease this problem.
- e. Isolated sandbox environment helps in more security
- f. A containerized application can be launched in minimal time.
- g. Managing library version dependency is a major concern, and containers ease this problem.
- h. Containers have helped in micro-services architecture.
- i. Containers can not be used where applications run close to the hardware
- j. Containers offer more portability, efficiency

Your answer is partially correct.

You have correctly selected 9.

The correct answers are: Managing library version dependency is a major concern, and containers ease this problem., Managing library version dependency is a major concern, and containers ease this problem., Changes in a shared library could break applications, but containers simplify upgrade and rollback., Isolated sandbox environment helps in more security, Isolated sandbox environment helps in more reliability and uptime, Containers offer more portability, efficiency, Containers offer more elasticity, reusability, Containers can not be used where applications run close to the hardware, Containers have helped in micro-services architecture., A containerized application can be launched in minimal time.

**Question 18**

Complete

Mark 1.00 out of 2.00

Write a shell program which does the following:

Takes two file-names as arguments on the command line.

Reads from the first file a pair of characters where the first is the "find" character and the second is the "replace" character.

Replaces all "find" characters in the second file with the "replace" character.

**For example,**

if the first file contains:

a m

[ ]

j t

Then the program will replace all "a" by "m" in the second file, all "[" by "]" in the second file, etc.

```
#!/bin/bash
while IFS=' ' read -r c1 c2
do
    sed -i "s/$c1/$c2/g" "$2"
done < $1
```

Comment:

**Question 19**

Complete

Mark 0.80 out of 1.00

Match the pairs

Web Conferencing	SaaS
VM in Cloud	PaaS
Ubuntu Machine in AWS	PaaS
Google Docs	SaaS
Load Balancer	IaaS

Your answer is partially correct.

You have correctly selected 4.

The correct answer is: Web Conferencing → SaaS, VM in Cloud → IaaS, Ubuntu Machine in AWS → PaaS, Google Docs → SaaS, Load Balancer → IaaS

**Question 20**

Complete

Mark 2.00 out of 2.00

Which of the following are NOT features of kubernetes?

- a. is portable
- b. automates software deployment
- c. is open source
- d. written in GO
- e. supports only docker
- f. supports declarative configuration
- g. written in C
- h. is extensible

The correct answers are: supports only docker, written in C

**Question 21**

Complete

Mark 1.00 out of 1.00

Given below are few statements differentiating between and comparing containers and virtual machines.

For each statement, mark True/False.

---

<b>True</b>	<b>False</b>
-------------	--------------

---

Example of container runtime is "dockerd", and an example of hypervisor is "kvm"

---

Container runtime and hypervisor do the same job

---

Virtual Machines need more storage compared to containers doing the same job

---

Virtual Machine runs its own kernel, but Container uses the kernel of the Host operating system

---

Example of container runtime is "docker", and an example of hypervisor is "secureboot"

---

Example of container runtime is "dockerd", and an example of hypervisor is "kvm": True

Container runtime and hypervisor do the same job: False

Virtual Machines need more storage compared to containers doing the same job: True

Virtual Machine runs its own kernel, but Container uses the kernel of the Host operating system: True

Example of container runtime is "docker", and an example of hypervisor is "secureboot": False

**Question 22**

Complete

Mark 1.00 out of 1.00

Match each docker command with its meaning.

docker build -t new .

Build a new docker image using Dockerfile in current directory, and tag it as "new"

docker images

show list of docker images available on the local machine

docker port

517065f6ab04

show all port mappings for the container with id 517065f6ab04

docker run --rm ubuntu

run the ubuntu image, and remove it when its done, as a result "docker ps -a" will show nothing here

Your answer is correct.

The correct answer is: docker build -t new . → Build a new docker image using Dockerfile in current directory, and tag it as "new", docker images → show list of docker images available on the local machine, docker port 517065f6ab04 → show all port mappings for the container with id 517065f6ab04, docker run --rm ubuntu → run the ubuntu image, and remove it when its done, as a result "docker ps -a" will show nothing here

**Question 23**

Complete

Mark 1.00 out of 1.00

Given below is a list of possible "values" for the "Kind" tag in the Kubernetes YAML syntax.

Select "Yes" if the said string can be used, and "No" if it is not a valid string.

(e.g. refer to the files like this

```
apiVersion: apps/v1
kind: Deployment
)
```

Yes	No
<input checked="" type="radio"/>	<input type="radio"/>

ReplicaSet

Service

PersistentVolume

Deployment

PersistentVolumeClaim

ReplicaSet: Yes

Pod: Yes

Service: Yes

PersistentVolume: Yes

Deployment: Yes

PersistentVolumeClaim: Yes

**Question 24**

Complete

Mark 2.00 out of 2.00

The command

```
docker run --rm -ti -v /root/data:/data:z --name fedora-1 fedora bash
```

```
docker run --rm -ti --volumes-from fedora-1 --name fedora-2 fedora
```

means

**True**    **False**

- One docker container of the fedora image is created, namely fedora-1 , and fedora-2 is an alias for that instance
- Running `ls /root/data` on host, `ls /data` on fedora1 and `ls /data` on fedora2 will show same result
- Two docker containers of the fedora image are created, namely fedora-1 and fedora-2
- Running `ls /var/lib/docker/volumes/root/data` on host, `ls /data` on fedora1 and `ls /data` on fedora2 will show same result
- The commands basically say that whatever is the volume for fedora1, should be shared with fedora2
- The `/root/data` folder on the Host is shared with the container(s)
- The commands basically say that whatever is the volume for fedora1, should be copied ditto with fedora2

One docker container of the fedora image is created, namely fedora-1 , and fedora-2 is an alias for that instance: False

Running `ls /root/data` on host, `ls /data` on fedora1 and `ls /data` on fedora2 will show same result: True

Two docker containers of the fedora image are created, namely fedora-1 and fedora-2: True

Running `ls /var/lib/docker/volumes/root/data` on host, `ls /data` on fedora1 and `ls /data` on fedora2 will show same result: False

The commands basically say that whatever is the volume for fedora1, should be shared with fedora2: True

The `/root/data` folder on the Host is shared with the container(s): True

The commands basically say that whatever is the volume for fedora1, should be copied ditto with fedora2: False

**Question 25**

Complete

Mark 1.00 out of 1.00

Why is "kubectl edit" not recommended, against "kubectl apply"?

- a. because it does change a running pod, and it may crash the pod
- b. because it changes a running pod
- c. because it does change a running pod, and no record of changes is maintained
- d. because it edits the YML file but does not apply it
- e. because it changes the pod when it is created next time

The correct answer is: because it does change a running pod, and no record of changes is maintained

**Question 26**

Complete

Mark 1.00 out of 1.00

Select from RHS what is Unique only to the Cloud Service mentioned on the LHS

SaaS	Application
PaaS	Operating System
IaaS	Networking

Your answer is correct.

The correct answer is: SaaS → Application, PaaS → Operating System, IaaS → Networking

◀ (Assignment) Kubernetes - 1

Jump to...

(Assignment) Kubernetes - 2 ►