ref: https://link.medium.com/Nfff0WJVI6

1. The entire cluster is broken. Investigate and fix the issues to bring back the cluster to normal healthy state. (Hint: you can ssh to master-00 or node-01). [4%]

   Tharindu's exam comment : same question. Different is needed to create manifests folder in the /etc/kubernetes path

   Answer

   Cluster and kubectl didn't work.

   Steps

   A. ssh to master node and run docker ps -a to figure out whether the control plane components are working.in this case there weren't any docker containers running in the master node. So I had to check the logs of the kubelet.
   B. Run the Command systemctl status kubelet.service. Check the logs and try to find out the issue. errors,
      -> "connection refused http://127.0.0.1:6443, master node could not be found".
      -> "invalid configurations, couldn't find the file path BROKEN ".
   C. Open the kubelet config file and check the configurations(you can find the config.yaml file location in the logs -> default location: `/var/lib/kubelet/config.yaml`).

The error was the invalid file location in staticPodPath: BROKEN. Check where the static pods are located in the master node. Normally the location is /etc/kubernetes/manifests. Check whether the control plane yaml files are there and fix the kubelet config file with the correct file location and save it. Restart the services systemctl daemon-reload and systemctl restart kubelet.service.

D. docker ps -a and check the control plane containers are running. Even though the pods were up running still kubectl didn't work. Again check the kubelet service logs -> systemctl status kubelet.service. Errors,

-> "connection refused http://127.0.0.1:6443".

E. I checked the kube-api-server pod yaml file in the location `/var/lib/kubelet/config.yaml` but coudn't find any issue there and again checked the ectd pod yaml. There I saw the container port was wrong and it was 2381 and I changed it to 2379. Restart the services systemctl daemon-reload and systemctl restart kubelet.service.

F. run some kubectl commands and check whether the cluster is working. Check all control plan nodes are working. Kubectl get pods -n kube-system.

2. The worker node-01 is not in a ready state. Investigate the problem and fix it. [I think 4%]

Tharindu's exam comment : same question.

ssh into the node-01 and check the kubelet status systemctl status kubelet.service. In my case it was dead. I just restarted the service and it made the cluster back to normal. systemctl restart kubelet.service.

3. Create a static pod name:app with image: nginx in the node-01. File location is /etc/kubernetes/manifests.

Ref: https://kubernetes.io/docs/tasks/configure-pod-container/static-pod/

Tharindu's exam comment : same question.

In this case the kubelet config didn't have the static pod path configuration. I had to add that path there and made the kubelet restart. Then Checked the pods are up and running in the correct node.

4. Run a daemonset name: my-daemon and image nginx. [3%]

Tharindu's exam comment : same question.

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: my-daemon
  labels:
    k8s-app:  my-daemon
spec:
  selector:
    matchLabels:
      name: my-nginx-daemon
  template:
    metadata:
```

```
    labels:
      name: my-nginx-daemon
  spec:
    tolerations:
    # this toleration is to have the daemonset runnable on master nodes
    # remove it if your masters can't run pods
    - key: node-role.kubernetes.io/master
      effect: NoSchedule
    containers:
    - name: my-daemon
      image: nginx
```

5. Find out all node names that are not with taint effect `NoSchedule` and save the result in the given file location.

Tharindu's exam comment : didn't get this question.

Easy way

```
#kubectl get node -o custom-columns=NAME:.metadata.name,TAINT:.spec.taints[*].effect |
grep -v NoSchedule
```

Hard way

```
kubectl get nodes  -o jsonpath='{"\n"}{range .items[*]}{.spec.taints}{" "}{.metadata.name}{"\n"}' |
grep -v NoSchedule
```

6. Find out the most CPU consuming pods with a given pod label and save the result in the given file location.

Tharindu's exam comment : same question. Different is only save most cpu consuming pod name

#kubectl top pods -n <namespace> --sort-by=cpu doesn't seem to work on 1.16 however you could use #kubectl top pods -n <namespace> | sort --reverse --key 2 --numeric

Checked with Binura and higher versions of 1.16.x worked with --sort-by=cpu

7. Create a deployment name: deployment-230 with image redis and expose it. Check the service and pod DNS with nslookup and save the result in a given 2 different files . [9% I think]

Tharindu's exam comment : same question.

```
#kubectl run deployment-230 --image=redis --restart=Always
#kubectl expose deployment deployment-230 --port=80 --target-port=8080
--type=NodePort
#kubectl run busybox --image=busybox:1.27 --restart=Never -- sh -c "sleep 3600"
#kubectl exec busybox -- nslookup deployment-230
```

#kubectl exec busybox -- nslookup 10-244-2-4.default.pod.cluster.local

8. Create a PV and PVC with given details. (access type, capacity, name). [4% I think]

Tharindu's exam comment : same question. Different is only create PV with hostpath

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: task-pv-volume
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 500Mi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: /mnt/data
```

```
PVC:
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: task-pv-claim
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
```

9. There is a pod yaml file and we have to add an init-container with **emptyDir** volume to create a .txt file in a given mount path. [7%]

Tharindu's exam comment : same question.

```
apiVersion: v1
kind: Pod
metadata:
  name: test-pd
spec:
  containers:
  - name: myapp-container
    image: busybox
    command: ['sh', '-c', 'echo The app is running! && sleep 3600']
    volumeMounts:
    - mountPath: /cache
      name: cache-volume
  initContainers:
  - name: init-touch-file
    image: busybox
    volumeMounts:
    - mountPath: /data
      name: cache-volume
    command: ['sh', '-c', 'touch > /data/test.txt']
  volumes:
  - name: cache-volume
    emptyDir: {}
```

10. There is a service and we have to find out all the pod names which implement that service. Don't use other methods and use the kubectl command to extract the data. (first check the file structure and labels -> `kubectl get svc -o jsonpath='{.items[*].spec.selector.name}'`).

Tharindu's exam comment : same question. Different , can be use any methods to get data (didn't said use any specific method)

```
#kubectl get svc <svc name> -o yaml {get the selector and then use that in the following}
#kubectl get pods --selector <label that is used in the svc> OR #kubectl get pods -l key=value
```

11. Create a deployment with name: deploy-003 and image nginx with version 1.10. Change the image version to 1.12 and again roll back to the previous version with all history records.[4%]

Tharindu's exam comment : same question. Different is no history record

```
#kubectl run deploy-003 --image=nginx:1.10 --record
#kubectl rollout history deployment/deploy-003 --revision=2 to check revision history of 2
#kubectl rollout undo deployment/deploy-003
```

12. Create a pod with images nginx + redis. Containers should be between 1 and 4. (multi containers pod). [4%]

Tharindu's exam comment : same question. Different is redis + nginx + memcached + consul

13. Create a pod with a new namespace. (pod name: pod-ns-kuu4, image: nginx, namespace: kuu4 will be provided).

Tharindu's exam comment : same question.

```
#kubectl create ns kuu4
#kubectl run nginx --image=nginx --restart=Never -n kuu4
#kubectl get pods -n kuu4 {to check if they are all created okay}
```

14. Create a secret with given key values. Then create two pods, one pod for Consuming Secret values from volumes with mount path and another pod for mount as an environment variable. Environment variables names will be given. [8-9% I think]

Tharindu's exam comment : same question. This is very important question it's 9 mark question

Create the secrets:

```
#kubectl create secret generic my-secret-stuff  --from-literal=key1=supersecret
--from-literal=key2=topsecret
```

Volume mounted

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod-sec1
```

```
spec:
  containers:
  - name: mypod-sec1
    image: busybox
    command:
    -  sleep
    -  "3600"
    volumeMounts:
    - name: foo
      mountPath: "/etc/foo"
      readOnly: true
  volumes:
  - name: foo
    secret:
      secretName: my-secret-stuff
```

Environment Variable:

```
apiVersion: v1
kind: Pod
metadata:
  name: secret-env-pod
spec:
  containers:
  - name: secret-env-pod
    image: busybox
    command:
    - sleep
    - "3600"
    env:
      - name: SECRET_KEY1
        valueFrom:
          secretKeyRef:
            name: my-secret-stuff
            key: key1
      - name: SECRET_KEY2
        valueFrom:
          secretKeyRef:
            name: my-secret-stuff
            key: key2
  restartPolicy: Never
```

15. Sort the PVs by capacity and save in a file. kubectl get pv --sort-by=.spec.capacity.storage [3%]

Tharindu's exam comment : same question.

16. There was a pod and svc already created but the pod couldn't access via the service. Need to find out the problem and fix it. (problem was the invalid target port set in the service).

Tharindu's exam comment : didn't get this question

17. There is a pod yaml file and we need to create that pod inside a specific given node. (have to set the `nodeSelector` field) .[2%]

Tharindu's exam comment : same question. (node selector has like disk=ssd)

```
#kubectl label node worker-2-k8s disk=ssd

apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: busybox
  name: busybox
spec:
  containers:
  - image: busybox
    name: busybox
    command:
    - sleep
    - "3600"
    resources: {}
  nodeSelector:
    disk: ssd
  dnsPolicy: ClusterFirst
  restartPolicy: Never
status: {}
```

18. Node cordon and drain. Asking to make one node for maintenance and another node for only unschedulable.

Tharindu's exam comment : same question. For only one node, set node as unschedulable and unavailable. Need pods scheduled in another node

19. They will give one master node and two worker nodes to create kubernetes Cluster from scratch. Tool was kubeadm.

Tharindu's exam comment : like same question.

20. There is pod, Check logs and extract log type ERROR and save it given specific file
#kubectl logs <podname> | grep "error string" [5%]

21. Scale given deployment to 6 replicas
#kubectl scale deployment deployment-name --replicas=6

22. Create a POD with given specifications with non persistence volume

23. Backup of ETCD
In the exam it is most likely etcdctl will already be installed, however if not installed you will need to install it. Please make sure that you install version 3.x.x as the version 2.x.x does not support the following command and method. I made a mistake on running sudo apt-get install etcd on Ubuntu 16.04 LTS and the packaging for 16.04 is 2.x.x.

Therefore if you are installing on Ubuntu 16.04 I recommend creating a bash script to install the latest 3.x version.

```
#!/bin/bash

ETCD_VERSION=${ETCD_VERSION:-v3.3.15}

curl -L
https://github.com/coreos/etcd/releases/download/$ETCD_VERSION/etcd-$ETCD_VERSION-li
nux-amd64.tar.gz -o etcd-$ETCD_VERSION-linux-amd64.tar.gz

tar xzvf etcd-$ETCD_VERSION-linux-amd64.tar.gz
rm etcd-$ETCD_VERSION-linux-amd64.tar.gz

cd etcd-$ETCD_VERSION-linux-amd64
sudo cp etcd /usr/local/bin/
sudo cp etcdctl /usr/local/bin/
```

```
rm -rf etcd-$ETCD_VERSION-linux-amd64
```

```
etcdctl --version
```

Then run the following command:

```
#sudo ETCDCTL_API=3 etcdctl --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert
/etc/kubernetes/pki/etcd/server.crt --key /etc/kubernetes/pki/etcd/server.key --endpoints
https://127.0.0.1:2379 snapshot save snapshot-backup
```

24. Create a service for an existing pod type NodePort [4%]
25. Create a deployment with 3 x replicas and save the file and get rid of the deployments etc and only save the json or the yaml file [3%]