Troubleshooting

Weight : 4

1) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`nginx-deployment` deployment pod not running, fix that issue

Solution:-

Step 1: edit deployment

```
kubectl edit deploy nginx-deployment
```

Step 2: Update From-

```
      initContainers:
      - command:
        - shell
        - echo 'Welcome To KillerCoda!'
```

To-

```
      initContainers:
      - command:
        - sh
        - -c
        - echo 'Welcome To KillerCoda!'
```

Step 3: Update From-

```
      volumes:
        - name: nginx-config
          configMap:
            name: nginx-configuration
```

To-

```
      volumes:
        - name: nginx-config
          configMap:
            name: nginx-configmap
```

Weight : 2

2) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`hello-kubernetes` pod not running, fix that issue

Solution:-

Step 1: edit pod

```
kubectl edit pod hello-kubernetes
```

Step 2: Update From-

```
  containers:
  - command:
    - shell
    - -c
    - while true; do echo 'Hello Kubernetes'; sleep 5; done
```

To-

```
  containers:
  - command:
    - sh
    - -c
    - while true; do echo 'Hello Kubernetes'; sleep 5; done
      initContainers:
```

Step 3: recreate new pod

```
kubectl replace -f /tmp/kubectl-edit-2019355827.yaml --force
```

Weight : 2

3) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`nginx-pod` pod not running, fix that issue

Solution:- Step 1: edit pod

```
kubectl edit pod nginx-pod
```

Step 2: Update From-

```
  - image: nginx:ltest
```

To-

```
  - image: nginx:latest
```

Weight : 8

4) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`redis-pod` pod not running, fix that issue

Solution:-

Step 1: describe pod

kubectl describe pod redis-pod
o/p:- Events:

Warning FailedScheduling 16s default-scheduler 0/2 nodes are available: persistentvolumeclaim "pvc-redis" not found. preemption: 0/2 nodes are available: 2 No preemption victims found for incoming pod..
"pvc-redis" not found so check the correct name of pvc
Step 2: kubectl get pvc

o/p:- NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

redis-pvc Pending manually 23s

"redis-pvc" is correct name so update pod
Step 3: kubectl edit pod redis-pod From- claimName: pvc-redis To- claimName: redis-pvc

And kubectl replace -f /tmp/kubectl-edit-2970798863.yaml --force

Still the pod is in pending state

Step 4: describe pod

kubectl describe pod redis-pod
o/p:- Events:

Warning FailedScheduling 13s default-scheduler 0/2 nodes are available: pod has unbound immediate PersistentVolumeClaims. preemption: 0/2 nodes are available: 2 No preemption victims found for incoming pod..

Check why "redis-pvc" is is unbound state
Step 5: describe pvc

kubectl describe pvc redis-pvc
o/p:- Events:

Warning ProvisioningFailed 14s (x12 over 2m58s) persistentvolume-controller storageclass.storage.k8s.io "manually" not found

We can observe here, given storage class name is “manually” instead of “manual” so update pvc
Step 6: kubectl edit pvc redis-pvc

From- storageClassName: manually To- storageClassName: manual

kubectl replace -f /tmp/kubectl-edit-2018407739.yaml --force

Check pvc status now, kubectl get pvc

o/p:- NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

redis-pvc Bound redis-pv 100Mi RWO manual 3s

Now it’s in Bound state, and check pod status now

Step 7: still pod is in pending state

kubectl describe pod redis-pod
o/p:- Events:

Warning Failed 14s kubelet Failed to pull image "redis:latested": rpc error: code = NotFound desc = failed to pull and unpack image "docker.io/library/redis:latested": failed to resolve reference "docker.io/library/redis:latested": docker.io/library/redis:latested: not found

We can observe here image name “redis:latested” instead “redis:latest”, so edit pod again

Step 8: kubectl edit pod redis-pod

From- image: redis:latested To- image: redis:latest

Step 9: Check the pod status now kubectl get pod, TADA Now it’s Running

---

Weight : 4

5) For this question, please set this context (In exam, diff cluster name)

```
kubectl config use-context kubernetes-admin@kubernetes
```

`frontend` pod is in Pending state, not running, fix that issue

Note: Don't remove any specification in `frontend` pod

Solution:-

Step 1: Let check why pod is in Pending state

kubectl describe pod frontend

o/p:- Events:

Warning FailedScheduling 18s default-scheduler 0/2 nodes are available: 1 node(s) didn't match Pod's node affinity/selector, 1 node(s) had untolerated taint {node-role.kubernetes.io/control-plane: }. preemption: 0/2 nodes are available: 2 Preemption is not helpful for scheduling..

Looks like node affinity or toleration configured on nodes, let’s check pod yaml

Step 2: kubectl get pod frontend -o yaml

Here, observer node affinity

```
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
        - matchExpressions:
            - key: NodeName
```

```
            operator: In
            values:
             - frontend
```

Step 3: now check node labels, `kubectl get nodes –show-labels`

O/p: - observe, labels `NodeName=frontendnodes` configured on node01 but in pod we saw key-value NodeName is frontend ,not frontendnodes so let's update pod

`kubectl edit pod redis-pod`

From- `- frontend` To- `- frontendnodes`

`kubectl replace -f /tmp/kubectl-edit-2018407739.yaml --force`

Check pod status now `kubectl get pod` now it's Running

Weight : 4

6) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`postgres-pod.yaml` is there, currently not able to deploy pod. check and fix that issue

Note: Don't remove any specification in `postgres-pod`

Solution:-

Step 1: replace From-

```
      tcpSocket:
        command:
          arg: 5432
```

To-

```
      tcpSocket:
        port: 5432
```

AND

```
    readinessProbe:
      exec:
        cmd:
```

To-

```
    readinessProbe:
      exec:
        command:
```

Weight : 5

7) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

something wrong in `redis-pod.yaml` pod template, fix that issue

Note: Don't remove any specification

Solution:-

Step 1: replace From-

```
    resources:
      requests:
        memory: "150Mi"
        cpu: "15m"
      limits:
        memory: "100Mi"
        cpu: "10m"
```

To-

```
    resources:
      requests:
        memory: "100Mi"
        cpu: "10m"
      limits:
        memory: "100Mi"
        cpu: "10m"
```

Run `kubectl apply -f redis-pod.yaml`

Weight : 4

8) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`my-pod-cka` pod is stuck in a Pending state, Fix this issue

Note: Don't remove any specification

Solution:-

Step 1: Check why Pod is in Pending state

`kubectl describe po my-pod-cka`
O/p:- `Events:`

Warning FailedScheduling 111s default-scheduler 0/2 nodes are available: pod has unbound immediate PersistentVolumeClaims. preemption: 0/2 nodes are available: 2 No preemption victims found for incoming pod..

Looks like pvc is in unbound state,

Step 2: Let's check pv and pvc

kubectl get pv,pvc
kubectl describe pv,pvc
pv is in ReadWriteOnce mode and pvc is in ReadWriteMany mode

Step 3: Let's edit pvc kubectl edit pvc my-pvc-cka

Replace From- - ReadWriteMany To- - ReadWriteOnce

And run kubectl replace -f /tmp/kubectl-edit-283826204.yaml --force

---

Weight : 4

9) For this question, please set this context (In exam, diff cluster name)

```
kubectl config use-context kubernetes-admin@kubernetes
```

just tainted node `node01` , update `tolerations` in this `application-deployment.yaml` pod template and create pod object

Note: Don't remove any specification

Solution:-

Step 1: Check taint on node01

kubectl describe node node01 | grep -i taint
Step 2: Update tolerations in application-deployment.yaml

```
  tolerations:
    - key: "nodeName"
      operator: "Equal"
      value: "workerNode01"
      effect: "NoSchedule"
```
Run kubectl apply -f application-deployment.yaml

Check pod status kubectl get pod

Weight : 2

10) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

some issue on the `controlplane` unable to run kubectl commands (EX: kubectl get node)

Solution:- Let's try once kubectl get node

Threw:- Unable to connect to the server: dial tcp: address 644333: invalid portbect

Looks like wrong port given in kubernetes config file, let edit

vi .kube/config

Replace From- https://172.30.1.2:644333 To- https://172.30.1.2:6443

try again kubectl get node

Weight : 5

11)For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`postgres-deployment.yaml` template is there, now we can't create object due to some issue in that, check and fix the issue

Note: Don't remove any specification

Solution:- looks like postgres-secret secret object is already available kubectl get secret

Update postgres-deployment.yaml From-

```
        - name: POSTGRES_USER
          valueFrom:
            secretKeyRef:
              name: postgres-secrte
              key: db_user
        - name: POSTGRES_PASSWORD
          valueFrom:
            secretKeyRef:
              name: postgres-secret
              key: db_password
```

To-

```
        - name: POSTGRES_USER
          valueFrom:
            secretKeyRef:
              name: postgres-secret
              key: username
        - name: POSTGRES_PASSWORD
```

```
            valueFrom:
              secretKeyRef:
                name: postgres-secret
                key: password
```

Weight : 4

12) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`nginx-pod` exposed to service `nginx-service` ,

when port-forwarded `kubectl port-forward svc/nginx-service 8080:80` it is stuck, so unable to access application `curl http://localhost:8080`

fix this issue

Solution:- labels not set to pod so add labels which is used in service `app: nginx-pod`

`kubectl edit po nginx-pod`

Add

```
  labels:

    app: nginx-pod
```

And try now `kubectl port-forward svc/nginx-service 8080:80` and `curl http://localhost:8080`

Weight : 4

13) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

In `controlplane` node, something problem with `kubelet` configuration files, fix that issue

You can `ssh controlplane`

Solution:-

Step 1: Check kubelet service running or not

`systemctl status kubelet.service`

It not running, looks like problem with configuration file only as mentioned in question

Step 2: Check /var/lib/kubelet/config.yaml

looks like problem with this

clientCAFile: /etc/kubernetes/pki/CA.CERTIFICATE
Change it to clientCAFile: /etc/kubernetes/pki/ca.crt

Step 3: Check /etc/kubernetes/kubelet.conf

Change it from server: https://172.30.1.2:64433333 to server: https://172.30.1.2:6443

Step 4: Use the following command to reload the kubelet service:

systemctl daemon-reload
Step 5: Restart the kubelet service: To ensure that the updated configurations take effect, restart the kubelet service:

systemctl restart kubelet.service

Check kubelet service status again

systemctl status kubelet.service

Weight : 2

14) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`stream-deployment` deployment is not up to date. observed `0` under the UP-TO-DATE it should be `1` , Troubleshoot, fix the issue and make sure deployment is up to date.

Solution:-

Step 1: Check deployment kubectl get deploy

Looks like scaled down to 0

Step 1: scale up to 1 kubectl scale deploy stream-deployment --replicas=1

Weight : 8

15) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`database-deployment` deployment pods are not running, fix that issue

Solution:-

Step 1: describe deployment pods to check reason `kubectl describe pod database-deployment-69799d647c-hsnsx`

O/p:- `Events:`

`Warning FailedScheduling 73s default-scheduler 0/2 nodes are available: persistentvolumeclaim "postgres-db-pvc" not found. preemption: 0/2 nodes are available: 2 No preemption victims found for incoming pod..`

Looks like “postgres-db-pvc” not there, check pvc `kubectl get pvc`

O/p:- `NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE`

`postgres-pvc Pending local-path 4m12s`

now we got correct pvc name “postgres-pvc” and we also observed it is in Pending state

Step 2: describe pvc to check reason `kubectl describe pvc postgres-pvc`

O/p:- `Normal WaitForFirstConsumer 5s (x24 over 5m50s) persistentvolume-controller waiting for first consumer to be created before binding`

Step 3: something problem with pvc, dig more get yaml of both pv and pvc

`kubectl get pv postgres-pv -o yaml`

`kubectl get pvc postgres-pvc -o yaml`

Yes, we got it in pv

```
accessModes:
- ReadWriteOnce
capacity:
  storage: 100Mi
```

But in pvc

```
accessModes:
- ReadWriteMany
resources:
  requests:
    storage: 150Mi
```

Step 3: lets correct pvc `kubectl edit pvc postgres-pvc`

From- `- ReadWriteMany` To- `- ReadWriteOnce`

From- `storage: 150Mi` To- `storage: 100Mi`

And run `kubectl replace -f /tmp/kubectl-edit-2231088049.yaml --force`

And now check pvc status `kubectl get pvc` its in Bound state

Step 4: lets edit deployment now From- `claimName: postgres-db-pvc` To- `claimName: postgres-pvc`

Check the pod status again `kubectl get pod` now it’s Running

Weight : 6

16) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`video-app` deployment replicas 0. fix this issue

expected: 2 replicas

Solution:- Step 1: check pod kubectl get pod no pods. check deployment kubectl get deploy

Step 2: let's describe deploy kubectl describe deploy video-app

O/p: Events: <none>

Step 3: looks like something problem with control plane components

kubectl get pods -A

kube-controller-manager-controlplane pod is in CrashLoopBackOff state lets dig more

kubectl describe pod kube-controller-manager-controlplane -n kube-system

O/p:- Warning Failed 21s (x4 over 79s) kubelet Error: failed to create containerd task: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: "kube-controller-manegaar": executable file not found in $PATH: unknown

Observered error:- exec: "kube-controller-manegaar" , it should be "kube-controller-manager"

Step 4: let's edit "kube-controller-manager" static pod yaml file

vi /etc/kubernetes/manifests/kube-controller-manager.yaml
Step 5: let's wait for some time both "kube-controller-manager" static pod and "video-app" deployment pods will come up

---

Weight : 4

17) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`red-pod` , `green-pod` , `blue-pod` pods are running, and `red-pod` exposed within the cluster using `red-service` service. and network policy applied on `red-pod` pod. problem is now the pod `red-pod` is accessible from both `green-pod` and `blue-pod` pods. fix the issue that `green-pod` only can able access `red-pod` pod.

Solution:- Step 1: check network policy yaml kubectl get netpol allow-green-and-blue -o yaml

```
spec:
  ingress:
```

```
  - from:
    - podSelector:
        matchLabels:
          run: green-pod
    - podSelector:
        matchLabels:
          run: blue-pod
```

Here, we can observe, this allowed traffic from both green-pod and blue-pod.

As per the question request, we need to remove traffic from blue-pod, remove below piece of code by running `kubectl edit netpol allow-green-and-blue`

```
    - podSelector:
        matchLabels:
          run: blue-pod
```

Weight : 4

18) For this question, please set this context (In exam, diff cluster name)

```
kubectl config use-context kubernetes-admin@kubernetes
```

`kubelet` service not running in `controlplane` , it will cause the `controlplane` in `NotReady` state, so fix this issue

Solution:- let's start the kubelet service: `systemctl start kubelet.service`

Check status now `systemctl status kubelet.service`

Weight : 5

19) For this question, please set this context (In exam, diff cluster name)

```
kubectl config use-context kubernetes-admin@kubernetes
```

when you run `kubectl get nodes` OR `kubectl get pod -A` threw :- `The connection to the server 172.30.1.2:6443 was refused - did you specify the right host or port?`

- need to wait for few seconds to make above command work again but above error will come again after few second

Expectation: `kube-apiserver-controlplane` pods running in `kube-system` namespace

You can `ssh controlplane`

Solution:-

Step 1: Let's try to run `kubectl get nodes`

O/p: `The connection to the server 172.30.1.2:6443 was refused - did you specify the right host or port?`

We know api server used to communication with cluster so something problem in API

Step 2: Let's check api static pod status `kubectl describe po kube-apiserver-controlplane -n kube-system`

O/p: `Warning Unhealthy 3m45s (x55 over 13m) kubelet Startup probe failed: Get "https://172.30.1.2:6433/livez": dial tcp 172.30.1.2:6433: connect: connection refused`

Observe here, probe port is not correct 6433, it should be 6443

Step 3: Let's update probe port in api static pod yaml `vi /etc/kubernetes/manifests/kube-apiserver.yaml`

```
    livenessProbe:
      failureThreshold: 8
      httpGet:
        host: 172.30.1.2
        path: /livez
        port: 6443
        scheme: HTTPS
      initialDelaySeconds: 10
      periodSeconds: 10
      timeoutSeconds: 15
    name: kube-apiserver
    readinessProbe:
      failureThreshold: 3
      httpGet:
        host: 172.30.1.2
        path: /readyz
        port: 6443
        scheme: HTTPS
      periodSeconds: 1
      timeoutSeconds: 15
    startupProbe:
      failureThreshold: 24
      httpGet:
        host: 172.30.1.2
        path: /livez
        port: 6443
        scheme: HTTPS
      initialDelaySeconds: 10
      periodSeconds: 10
      timeoutSeconds: 15
```

Step 4: wait for sometime(sometime more time), pod will come up

Weight : 4

20) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`postgres-deployment` deployment pods are not running, fix that issue

Solution:-

Step 1: describe deployment pods to check reason kubectl describe pod postgres-deployment-6cc57cb67b-lqg9d

O/p:- Events:

Warning Failed 4s kubelet Error: configmap "postgres-db-config" not found

Looks like “postgres-db-config” not there, check configmap kubectl get cm

O/p:- NAME DATA AGE

kube-root-ca.crt 1 17d

postgres-config 2 2m34s

now we got correct configmap name “postgres-config”

Step 2: edit deployment kubectl edit deploy postgres-deployment

To-

```
      - name: POSTGRES_DB
        valueFrom:
          configMapKeyRef:
            key: POSTGRES_DB
            name: postgres-config
      - name: POSTGRES_USER
        valueFrom:
          configMapKeyRef:
            key: POSTGRES_USER
            name: postgres-config
```

check deployment new pod again kubectl get pods

Looks like still not coming up, dig more kubectl describe po postgres-deployment-54dc976c54-56lxv

O/p:- Events:

Warning Failed 7s (x7 over 101s) kubelet Error: secret "postgres-db-secret" not found

Looks like “postgres-db-secret” not there, check secret kubectl get secret

O/p:- NAME TYPE DATA AGE

postgres-secret Opaque 1 8m27s

now we got correct configmap name “postgres-secret”

Step 3: again edit deployment kubectl edit deploy postgres-deployment

Check pod status now kubectl get pods , Yes Running now

Weight : 2

21) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`frontend-deployment.yaml` deployment template is there, try to deploy, if there is any issue fix that

Solution:- yaml looks fine let’s try to apply

kubectl apply -f frontend-deployment.yaml

O/p:- Error from server (NotFound): error when creating "frontend-deployment.yaml": namespaces "nginx-ns" not found

Looks like there;s no nginx-ns namespace, let create kubectl create ns nginx-ns

Try again kubectl apply -f frontend-deployment.yaml

Check pods status kubectl get po -n nginx-ns , Yes Running now

---

Weight : 4

22) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`my-pvc` Persistent Volume Claim is stuck in a Pending state, fix this issue, make sure it is in Bound state

Solution:- let’s check pv and pvc yaml

Step 1: let’s check pv and pvc yam

kubectl get pv my-pv -o yaml

kubectl get pvc my-pvc -o yaml

Yes, we got it in pv

accessModes:

```
  - ReadWriteOnce
  capacity:
    storage: 100Mi
```

But in pvc

```
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 150Mi
```

Step 2: lets correct pvc `kubectl edit pvc my-pvc`

From- `- ReadWriteMany` To- `- ReadWriteOnce`

From- `storage: 150Mi` To- `storage: 100Mi`

And run `kubectl replace -f /tmp/kubectl-edit-3333632057.yaml --force`

check pvc status now `kubectl get pvc` , Yes its Bound now

Weight : 6

23) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

`cka-pod` pod exposed internally within the service name `cka-service` and for `cka-pod` monitor(access through svc) purpose deployed `cka-cronjob` cronjob that run `every minute` .

Now `cka-cronjob` cronjob not working as expected, fix that issue

Solution:-

Step 1: first check whether svc accessing pod or not

`kubectl port-forward service/cka-service 8080:80`

Looks like stuck

Step 2: first check pod and service yaml

`kubectl get pod cka-pod -o yaml`

`kubectl get svc cka-service -o yaml`

Observe, label not added to pod that's why port-forward stuck, let update pod

`kubectl edit pod cka-pod`

Add this under metadata

```
  labels:
    app: cka-pod
```

Try again `kubectl port-forward service/cka-service 8080:80`

Yes its working now

Step 3: let's check cronjob yaml why its failing to monitor and it should run every minute

`kubectl get cronjobs cka-cronjob -o yaml`

`schedule: '* * * * *'` wrong schedule

Observe, accessing pod instead service

```
          containers:
          - command:
            - curl
            - cka-pod
```

Step 4: lets edit `kubectl edit cronjobs cka-cronjob`

To-

`schedule: '*/1 * * * *'` and

```
          containers:
          - command:
            - curl
            - cka-service
```

Weight : 4

24) For this question, please set this context (In exam, diff cluster name)

```
kubectl config use-context kubernetes-admin@kubernetes
```

You have a service account named `dev-sa` , a Role named `dev-role-cka` , and a RoleBinding named `dev-role-binding-cka` . we are trying to `create` `list` and `get` the `pods` and `services` . However, using `dev-sa` service account is not able to perform these operations. fix this issue.

Solution:- Step 1: check role yaml `kubectl get role dev-role-cka -o yaml`

Now permission given to

```
    resources:
    - secrets
    verbs:
    - get
```

Update to- `kubectl edit role dev-role-cka`

```
 resources:
 - pods
 - services
 verbs:
 - get
 - create
 - list
```

Weight : 4

25) For this question, please set this context (In exam, diff cluster name)

```
kubectl config use-context kubernetes-admin@kubernetes
```

You have a service account named `prod-sa` , a Role named `prod-role-cka` , and a RoleBinding named `prod-role-binding-cka` . we are trying to `create list` and `get` the `services` . However, using `prod-sa` service account is not able to perform these operations. fix this issue.

Solution:- Step 1: check role yaml `kubectl get role prod-role-cka -o yaml`

Now permission given to

```
  resources:
  - pods
  verbs:
  - list
```

Update to- `kubectl edit role prod-role-cka`

```
 resources:
 - services
 verbs:
 - get
 - create
 - list
```

Weight : 5

26) For this question, please set this context (In exam, diff cluster name)

```
kubectl config use-context kubernetes-admin@kubernetes
```

`cache-daemonset` DaemonSet deployed, now it's not creating any pod on the `controlplane` node. fix this issue and make sure the pods are getting created on all nodes including the controlplane node as well.

Solution:-

Step 1: check pods are assigned which node

kubectl get po -o wide | grep cache-daemonset

O/p:- cache-daemonset-fhdmq 1/1 Running 0 52s 192.168.1.3 node01 <none> <none>

Pods in only node01

Step 2: let get taint on controlplane node and add toleration to daemonset

kubectl describe node controlplane | grep -i taint

Step 3: let edit daemonset kubectl edit ds cache-daemonset

Add this under container section

```
      tolerations:
        - key: node-role.kubernetes.io/control-plane
          effect: NoSchedule
```

Check pod again kubectl get pods -o wide

Weight : 5

27) For this question, please set this context (In exam, diff cluster name)

`kubectl config use-context kubernetes-admin@kubernetes`

something is not working at the moment on controlplane node(Cause NotReady state), check that and `etcd-controlplane` pod is running in `kube-system` environment, take backup and store it in `/opt/cluster_backup.db` file, and also store backup console output store it in `backup.txt`

`ssh controlplane`

Solution:-

Step 1: check kubelet service status systemctl status kubelet.service

Not running so systemctl start kubelet.service

Step 1: Take backup

etcdctl --endpoints=https://127.0.0.1:2379 --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/server.crt --key=/etc/kubernetes/pki/etcd/server.key snapshot save /opt/cluster_backup.db

Step 2: Save console o/p in a file backup.txt