CKA exam Q01:
#Create Alias for kubectl command:
 alias g=kubectl
Q 01) #Create a new pod called admin-pod with image busybox. Allow it to be able to set system_time. Container should sleep for 3200 seconds.

Question 03) Create a new deployment called web-proj-268 with image nginx:1.16 and one replica. Next, upgrade the deployment to version 1.17 using rolling update.
Make sure that the version upgrade is recorded in the resource annotation.

Question 04) Create a new deployment web-003, scale this deployment to 3 replicas, make sure desired number of pods are always running.
Create exam scenario using below command :
 sudo sed -i 's/\- kube-controller-manager/\- kube-controller-man/g' /etc/kubernetes/manifests/kube-controller-manager.yaml
  #kubernetes #CKA #DevOps #security

Question 05) Upgrade given cluster (master and worker node) from 1.23.8-00 to 1.24.2-00. Make sure to first drain respective node prior to update
and make it available post update.
 #kubernetes #CKA #DevOps #security

Our cluster is running latest code so we need to First downgrade nodes for upgrade (use 3 step copy paste as instructed for downgrade)

STEP 1 On Node6:
############
echo '
alias g=kubectl
g drain node6 --ignore-daemonsets
g get pods -A -o wide
sudo apt update

```
#apt-cache madison kubeadm
sudo  apt -y install kubeadm=1.23.8-00 --allow-downgrades
sudo kubeadm  upgrade apply v1.23.8 -y
sudo apt install kubelet=1.23.8-00 --allow-downgrades -y
sudo systemctl restart kubelet
g uncordon node6
sleep 20
g drain node7 --ignore-daemonsets
g get pods -A -o wide' |tee control-node_grade.sh


source ./control-node_grade.sh
############
STEP 2)Wait for above script to finish and then, Come back to nodet, run it on
node7
############
echo '
sudo apt update
sudo  apt -y install kubeadm=1.23.8-00 --allow-downgrades
sudo kubeadm upgrade  node
sudo apt install kubelet=1.23.8-00 --allow-downgrades -y
sudo systemctl restart kubelet ' |tee node7-node_grade.sh


source ./node7-node_grade.sh
sudo reboot


############
STEP 3) Wait for above script to finish and then, Come back to node6 and run
below command (downgrade expects one reboot on each node):
############
g uncordon node7
g uncordon node6
sudo reboot


############
```

Q 6) deploy a web-load-5461 pod using nginx:1.17 with the label set to

tier=web

Q 7) Create static pod on node07 called static-nginx with image nginx and you have to make sure that it is recreated/restarted automatically in case of any failure happens.
It's easy:

Q 8) Create a pod called pod-multi with 2 containers as it is descripted below:
Container 1 : name:container1, image: nginx
Container 2 : name:container2, image: busybox, command: sleep 4800

Q 9) Create a pod called delta-pod in defence namespace belonging to the development environment (env=dev) and frontend tier (tier=front), image: nginx:1.17

Q10) Get web-load-5461 pod details in json format and store it in a file at /opt/output/web-load-5461-j070822n.json

Q11) Backup ETCD database and save it root with name of backup "etcd-backup.db"

Q12) A new application finance-audit-pod is deployed in finance namespace. Find out what is wrong with it and fix the issue.
NOTE: No configuration changes allowed, you can only delete or recreate the pod.
Below command will create a scenario for us:
g create ns finance ; g run finance-audit-pod --image=busybox -n finance --command sleeo 180

Q 13 :use JSONPath query to retrieve our OS images of all K8s nodes and store it in a file ~/allNodeOSImages8.txt

Q14) Create a persistent volume with given specifications:
Volume Name - pv-rnd
storage - 100Mi
Access modes - ReadWriteMany
host path - /pv/host-data-rnd

Q15) Expose "audit-web-app" pod to by creating a service "audit-web-app-service" on port 30002 on nodes of given cluster.
Note : Now given web application listens on port 8080
use below command to create an exam like scenario:
kubectl run audit-web-app --image=nginx --port=8080

Q16) Create a pod called pod-jxc, using details mentioned below:
SecurityContect

runasUser: 1000
fsGroup: 2000

Image=redis:alpine

It is critical to understand that taints and tolerations are only enforced at the node level, and pods have the freedom to choose nodes without taints, but if all of our nodes are tainted then new pods must be with exact tolerations defined in them.

Q17) Apply taint a worked node node7 with details provided below:
Create a pod called dev-pod-nginx using image=nginx,
make sure workloads are not scheduled to this worker node (node7)
Create another pod prod-pod-nginx using image=nginx with a toleration to be scheduled on node7.

Details :key:env_type, value:production, operator: Equal & effect: NoSchedule

Q18) Create a user “nec-adm". Grant nec-adm access to cluster, should have permissions to create, list, get, update, and delete pods in nec namespace
Private key exist in location: /vagrant/nec-adm.key and csr at /vagrant/nec-adm.csr

Create exam scenario using below commands:
g create ns nec
openssl genrsa -out nec-adm.key 2048
openssl req -new -key nec-adm.key -out nec-adm.csr

Q19) Craete a PersistentVolume, PersistentVolumeClaim and Pod with below specifications

PV - name : mypvl , Size: 100Mi, AccessModes: ReadWritemany, Hostpath: /pv/log, Reclaim Policy: Retain
PVC - name: pv-claim-l, Storage request: 50Mi, Access Modes:

ReadWritemany
Pod - name : my-nginx-pod, image Name: nginx, Volume: PersistentVolumeClaim: pv-claim-l, volume mount : /log

Q20) Worker node node7 is not responding, have a look and fix the issue

Create exam scenario run these 2 commands:
ssh node7 \ sudo sed -i 's/ca.crt/YOU_ARE_LOOKING_FOR_ME.crt/g' /var/lib/kubelet/config.yaml
ssh node7 \ sudo systemctl restart kubelet

21) List internal IPs of all nodes of given cluster, save result to a file /root/InternalIPList
Answer should be in a format: Internal IP of 1st Node (space) Internal IP od 2nd node (in a single line)

22) CKA exam Q22
Create a new deployment called nginx-deployment with an image nginx:1.16 and 5 replicas. There are 2 worker nodes in our cluster.
Please make sure no pod will get deployed on node7.

Note: Revert any changes that you do on this environment.

23) CKA exam Q23
Create a replicaset (name : web-replica, image=nginx, replicas=3), there is already a pod running in our cluster.
Please make sure that total count of pods running in the cluster is not more than 3.

Create given pod well in advance using below command:

g run web-critical --image=nginx --port 8080 --labels app=web

24) CKA exam Q24
We have worker 3 nodes in our cluster, create a DaemonSet (name prod-pod, image=nginx) on each node except worker node8.

25) CKA exam Q25
A pod “prod-pod” (image=nginx, port 8080) in default namespace is not running. We need fix it and bring it in running state.

26) CKA exam Q26
A pod “my-data-pod” in data namespace is not running. Fix the issue and get it in running state.

Note: All supported definition files are placed at root.

To create question scenario just change pv1claim.yaml and remove namespace information (ensure data namespace was created already) and apply them.

27) (NOTE: I am running this solution in a minikube environment)

```
g run web-pod --image=nginx
g expose pod web-pod --name=web-pod-svc --port 80
g get pods
g get pods -o wide
g describe svc web-pod-svc
g run nscheck --image=busybox:1.28 --command sleep 4800
g get pods -o wide
g exec -it nscheck -- nslookup web-pod-svc
g exec -it nscheck -- nslookup web-pod-svc |tee  web.svc
g exec -it nscheck -- nslookup 172-17-0-3.default.pod
g exec -it nscheck -- nslookup 172-17-0-3.default.pod | tee  web.pod
```

CKA exam Q28
We have deployed a pod called web-pod (where port 80 was exposed using service web-pod). Incoming connections (from a pod apicheck) to web-pod service are not working.
Identify and make certain changes on pod apicheck (re-deploy is needed) so that it can access web-pod.
Note: A NetworkPolicy named api-allow was also created as part of the deployment.

Run following commands to create exam like environment:

```
echo 'kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: api-allow
spec:
  podSelector:
    matchLabels:
```

```
      app: bookstore
      role: api
  ingress:
  - from:
      - podSelector:
          matchLabels:
            app: bookstore' |tee  api-allow.yaml


 g apply -f api-allow.yaml
  g run web-pod --image nginx --labels app=bookstore,role=api --expose --
port  80
 g run apicheck --image alpine  --labels role=api --command sleep 4800
```

31 - CKA exam Q31 with Solution... Kubernetes network policy

```
echo 'apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: internal-policy
  namespace: default
spec:
  podSelector:
    matchLabels:
      name: internal
  policyTypes:
  - Egress
  - Ingress
  ingress:
    - {}
  egress:
  - to:
    - podSelector:
        matchLabels:
          db: mysql
    ports:
```

```
    - protocol: TCP
      port: 3306

  - to:
    - podSelector:
        matchLabels:
          app: payroll
    ports:
    - protocol: TCP
      port: 8080

  - ports:
    - port: 53
      protocol: UDP
    - port: 53
      protocol: TCP' |tee internal-policy.yaml
```

33 - CKA exam Q33 with Solution... create a network policy for incoming web connections

Command to create question environment:

```
alias g=kubectl
g run web-test --image nginx
g expose pod  web-test --name web-test-svc --type NodePort --port 80
g run connect-pod --image busybox --command sleep 4800
g exec -it connect-pod -- wget

echo '---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: default-deny-ingress
spec:
  podSelector: {}
  policyTypes:
  - Ingress' |tee deny-all.yaml


g apply -f deny-all.yaml
```

37 - CKA exam Q37 ... mount secret in 2 pods using filesystem and environment variable

```
echo 'apiVersion: v1
kind: Pod
metadata:
  name: pod-sec-file
spec:
  containers:
  - image: redis
    name: redis
    volumeMounts:
    - name: sec1
      mountPath: "/secrets"
  volumes:
  - name: sec1
    secret:
     secretName: sec1' |tee  pod-sec-file.yaml
##
echo 'apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  name: pod-sec-env
spec:
  containers:
  - image: redis
    name: redis
    env:
    - name: CONFIDENTIAL
      valueFrom:
        secretKeyRef:
          name: sec1
          key: password' |tee  pod-sec-env.yaml
```

42 - CKA exam Q42 with Solution... list all workloads (resources) in a Kubernetes cluster

One should memorize below command :

#kubectl get ds,sts,deploy,rs,rc -A

where:
rs is for ReplicaSets
ds is DaemonSets
sts is for StatefulSets
deploy is for Deployments
rc is for ReplicationControllers