**Weight: 12**

Create a new service account with the name pvviewer. Grant this Service account access to list all PersistentVolumes in the cluster by creating an appropriate cluster role called pvviewer-role and ClusterRoleBinding called pvviewer-role-binding.  
Next, create a pod called pvviewer with the image: redis and serviceAccount: pvviewer in the default namespace.

* ServiceAccount: pvviewer
* ClusterRole: pvviewer-role
* ClusterRoleBinding: pvviewer-role-binding
* Pod: pvviewer
* Pod configured to use ServiceAccount pvviewer ?

**Weight: 12**

List the InternalIP of all nodes of the cluster. Save the result to a file /root/CKA/node\_ips.

Answer should be in the format: InternalIP of controlplane<space>InternalIP of node01 (in a single line)

* Task Completed

**Weight: 12**

Create a pod called multi-pod with two containers.  
Container 1, name: alpha, image: nginx  
Container 2: name: beta, image: busybox, command: sleep 4800  
  
Environment Variables:  
container 1:  
name: alpha  
  
Container 2:  
name: beta

* Pod Name: multi-pod
* Container 1: alpha
* Container 2: beta
* Container beta commands set correctly?
* Container 1 Environment Value Set
* Container 2 Environment Value Set

**Weight: 8**

Create a Pod called non-root-pod , image: redis:alpine  
  
runAsUser: 1000  
  
fsGroup: 2000

* Pod non-root-pod fsGroup configured
* Pod non-root-pod runAsUser configured

**Weight: 14**

We have deployed a new pod called np-test-1 and a service called np-test-service. Incoming connections to this service are not working. Troubleshoot and fix it.  
Create NetworkPolicy, by the name ingress-to-nptest that allows incoming connections to the service over port 80.

Important: Don't delete any current objects deployed.

* Important: Don't Alter Existing Objects!
* NetworkPolicy: Applied to All sources (Incoming traffic from all pods)?
* NetWorkPolicy: Correct Port?
* NetWorkPolicy: Applied to correct Pod?

**Weight: 12**

Taint the worker node node01 to be Unschedulable. Once done, create a pod called dev-redis, image redis:alpine, to ensure workloads are not scheduled to this worker node. Finally, create a new pod called prod-redis and image: redis:alpine with toleration to be scheduled on node01.

key: env\_type, value: production, operator: Equal and effect: NoSchedule

* Key = env\_type
* Value = production
* Effect = NoSchedule
* pod 'dev-redis' (no tolerations) is not scheduled on node01?
* Create a pod 'prod-redis' to run on node01

**Weight: 8**

Create a pod called hr-pod in hr namespace belonging to the production environment and frontend tier .  
image: redis:alpine

Use appropriate labels and create all the required objects if it does not exist in the system already.

* hr-pod labeled with environment production?
* hr-pod labeled with tier frontend?

**Weight: 8**

A kubeconfig file called super.kubeconfig has been created under /root/CKA. There is something wrong with the configuration. Troubleshoot and fix it.

* Fix /root/CKA/super.kubeconfig

**Weight: 14**

We have created a new deployment called nginx-deploy. scale the deployment to 3 replicas. Has the replica's increased? Troubleshoot the issue and fix it.

* deployment has 3 replicas