1. – Create a secret key1=value4; create a pod with nginx image which consumes the secret as env variable BEST\_VARIABLE.
2. – Create a pod with redis image in web namespace. Expose port 6379.
3. – Create a cronjob hello with container hello which runs every minute and should be terminated if not completed in 17 sec. Use busybox image and print date. Hint: Use the cronjob yaml file from the documentation and add activeDeadlineSeconds: 17 in pod spec
4. – Create a pod app1 with container appcont with args (no commands). Use declarative method. Once pod is created output summary in json format to a file.

Hint: Use kubectl run app1 –image=someimage –restart=Never --dry-run=client –o yaml -- -lines 56 –F > somefile.yaml (edit the name of container)

Kubectl get po app1 –o json > somefile.json

1. – Deployment already created in a namespace. Change replicas to 4 and add label. Once deployment is running, create a service which expose port 81 using the label selector you added above. Type=NodePort
2. – pod yaml file provided. Create a pod from that yaml file. Output logs to a file
3. – Deployment running in a namespace. Edit the deployment to use serviceaccount account1. Sa already created. Hint: Use kubectl set sa deployment nameofdeployment account1
4. – Create a pod which requests certain amount of resources. 2Gi memory and 400m cpu. Use nginx image.
5. – Create a configmap some-configmap using key1/valu4. Create a pod using nginx image which mounts the configmap as volume mount
6. – Deployment/pod broken. Check the image name and edit (nginx spelled incorrectly)
7. – Find the pod consuming most cpu and output that pod’s name to a text file. Namespace given
8. – Broken deployment. Liveness probe failing. First, found which pod is broken in what namespace (4 ns). Output the broken pod in a file in namespace/pod format. Fix the issue.
9. – Create a deployment with nginx image with env variable NGINX\_PORT=8000. Hint: Create the deployment with declarative/imperative method and then add the env variable using command kubectl set env deployment nameofdep NGINX\_PORT=8000
10. – Change the maxsurge=4 and maxunavailable=2 in a running deployment. Update the image to 1.18.0 and then rollback.
11. – Two deployments i.e. api & proxy exposed to 2 different services running. A pod web-gateway is deployed but is unable to communicate with services. Edit the pod so that it can talk with two services. Hint: Get networkpolicies i.e. kubectl get networkpolicies then examine the networkpolicies using kubectl describe networkpolicies api… && kubectl describe networkpolicies proxy… Find the label in ingress section should be something like api-access is true & proxy-access=true. Edit the web-gateway pod and add labels api-access: “true” & proxy-access: “true”.
12. – Create index.html file on a node and add some text TEXT=ABC (you have to ssh to the worker node create file in directory e.g. /var/lib and then return to base node). Create pv using hostPath (dir above i.e. /var/lib) which allocate 1 Gi storage. ReadWriteOnce. StorageClass name is given. Create pvc with 200 Mi volume with ReadWriteOnce access mode. Create a pod (image not specified) with label app=storage which mounts the volume on mountPath /usr/share…….
13. – pod running on cluster but not responding. Two endpoints i.e. /started and /healthz. The pod accepts traffic if HTTP 200 is sent and will not accept traffic with HTTP 500. Configure the pod with those endpoints using port 8080. /started=readinessProbe & /healthz=livenessprobe.
14. – sidecar question. Create deployment which will convert logs from format A to format B. Step 1 – Create configmap with yaml file provided and then create a multicontainer deployment. Specify some commands (provided) on 1st container and mount /tmp/log on both containers. Mount configMap as volume on 2nd sidecar container. /tmp/log should be non-persistent so use volume emptyDir: {}.
15. – Ambassador pattern question. Service already running which is exposed on port 60 but it doesn’t respond when port 9090 is used. First create a configMap with provided config file. Edit multicontainer pod spec file (given) and change command section on 1st container from nginxsvc to localhost. Add mountpath on 2nd container of type configMap.