Kubernetes Exam preparation question:

1. View top CPU consuming pods using **kubectl top** command. And **store output to a file.**
2. **Sort pods** according to **creationTimestamp.**
3. View **pods** associated with a service(example nginx service). Remember service and pods are different.
4. Use **secrets** as **environment variables**.
5. Create an **RBAC** for **adding new node** to the cluster.
6. Create an **init container,** which creates an empty file inside pod.(hint: you have to use shell command to create a file **# echo > file\_name**)
7. **Expose service** and **create endpoints**.
8. Use configMaps to set environment variable(this might be tricky question, do not get confused with the terms provided in question **ie. Key and values**)
9. Create **deployment**.
10. Create **pods with container name** provided in question(I mean question in exam).
11. Run **pods under particular namespace** (will be provided in exam).
12. Mount **volume** **as non-persistent**.
13. Create **cronjob** to print **“hello world”.**
14. **Scaling up deployments**.
15. **Rolling updates** to upgrade nginx version.
16. Use **rollback**.
17. **Reschedule** all the pods from one node to another. Make node(name mentioned in question) non-schedulable.
18. Create **pod with two containers** one will be nginx and another should be redis.
19. Create **resource limit for namespaces**. (CPU and RAM limit) and run pods inside it.
20. **List pods using curl** command(remember api access lab (auth, client cert, ca-cert))
21. Use **proxy**
22. Create **DaemonSets.**
23. **Schedule deployment to run pods only on a particular node.**
24. Expose **Deployment on Nodeport / ClusterIP**
25. Taint and tolerance.
26. Using TLS for **Authentication and authorization**.
27. Create RBAC specified for the **users and roles** in the question(create, delete , get etc…)

21-Aug-2019:

**Short questions:**

1. Create **DaemonSets (use nginx)** and pod name provided in question**.**
2. Create **init** container which create empty file inside pod.
3. A manifest file is placed in a path, update the manifest to create init container and make sure init container starts and then the container provide in the manifest.

1. **Sort Pods** : creationTimestamp, name, cpu/mem utilization and store output in a file( path mentioned in exam)
2. Create **pod with two containers** one will be nginx and another should be redis(Memcached).
3. **Scaling up deployments**.
4. **Reschedule** all the pods from one node to another. Make node(name mentioned in question) non-schedulable (drain).
5. Taint and tolerance question.
6. Sort a given PV as per storage using Kubectl only and store output to file.
7. Check the log of the given container and paste the lines which contains errors (Error will be provided in exam) to a file.
8. Create a deployment with given values (Metadata like name, API version etc) with nginx image version 1.11, update the container with image nginx 1.12 and rollback deployment.
9. Create a service with node port for a pod with label app=web.
10. List the nodes which are not in ready state (Those shouldn’t be tainted).
11. Find the pods be given label and find out pod with highest CPU percentage.
12. Scale the deployment to 5 replicas (deployment name given in question).

**Moderate Questions:**

**Volumes**

1. Create secret as below:

**Name: super-secret**

**Credentials: bob**

Create a pod with image nginx and mount the secret as volume in (path will be given in question),

1. Create a pod with image ngnix and name pod-env and export credentials as CREDENTIAL.
2. Create a PV named <name will be given> using hosted path.
3. Create pod named dummy which uses non-persistent volume in a particular namespace.

Node:

1. Create a new node and add to the cluster.
2. Investigate the node in not ready state and fix it.
3. Investigate the problem in the cluster, check the services and fix them.

**Lengthy Questions:**

1. Create an **RBAC** for **adding new node** to the cluster.
2. Create RBAC specified for the **users and roles** in the question (create, delete , get etc…)
3. Using TLS for **Authentication and authorization.**
4. ETCD Backup --- endpoint is given, certificate path are given, backup path is also given.
5. Create a deployment with dummy image nginx and expose the service, use the busybox:1.28 image to nslookup the created service and put the output in a file called service.dns, nslookup the pod created above and put the output value in the file called pod.dns.