My Exam Prep:

* CNCF Kubernetes Class + labs
* K8S The Hard Way run through
* Run through all the tasks in the k8s docs
* Practice with systemd, journald, openssl, cfssl, and etcd
* Work through the sections in Walid’s github list

Try the following exercises interactively:

Note - there are no answers here on purpose. You should be able to do these yourself using the minimal docs that you are allowed to use during the test. At a minimum this should train you on where to look for this info during the test, without notes.

1. Create a node that has a SSD and label it as such.
2. Create a pod that is only scheduled on SSD nodes.
3. Create 2 pod definitions: the second pod should be scheduled to run anywhere the first pod is running - 2nd pod runs alongside the first pod.
4. Create a deployment running nginx version 1.12.2 that will run in 2 pods  
    a. Scale this to 4 pods.  
    b. Scale it back to 2 pods.  
    c. Upgrade this to 1.13.8  
    d. Check the status of the upgrade  
    e. How do you do this in a way that you can see history of what happened?  
    f. Undo the upgrade
5. Create a service that uses a scratch disk.  
    a. Change the service to mount a disk from the host.  
    b. Change the service to mount a persistent volume.
6. Create a pod that has a liveness check
7. Create a service that manually requires endpoint creation - and create that too
8. Create a daemon set  
    a. Change the update strategy to do a rolling update but delaying 30 seconds between pod updates
9. Create a static pod
10. Create a busybox container without a manifest. Then edit the manifest.
11. Create a pod that uses secrets  
     a. Pull secrets from environment variables  
     b. Pull secrets from a volume  
     c. Dump the secrets out via kubectl to show it worked
12. Create a job that runs every 3 minutes and prints out the current time.
13. Create a job that runs 20 times, 5 containers at a time, and prints "Hello parallel world"
14. Create a service that uses an external load balancer and points to a 3 pod cluster running nginx.
15. Create a horizontal autoscaling group that starts with 2 pods and scales when CPU usage is over 50%.
16. Create a custom resource definition  
     a. Display it in the API with curl
17. Create a networking policy such that only pods with the label access=granted can talk to it.  
     a. Create an nginx pod and attach this policy to it.   
     b. Create a busybox pod and attempt to talk to nginx - should be blocked  
     c. Attach the label to busybox and try again - should be allowed
18. Create a service that references an externalname.  
     a. Test that this works from another pod
19. Create a pod that runs all processes as user 1000.
20. Create a namespace  
     a. Run a pod in the new namespace  
     b. Put memory limits on the namespace  
     c. Limit pods to 2 persistent volumes in this namespace
21. Write an ingress rule that redirects calls to /foo to one service and to /bar to another
22. Write a service that exposes nginx on a nodeport  
     a. Change it to use a cluster port  
     b. Scale the service  
     c. Change it to use an external IP  
     d. Change it to use a load balancer
23. Deploy nginx with 3 replicas and then expose a port  
     a. Use port forwarding to talk to a specific port
24. Make an API call using CURL and proper certs
25. Upgrade a cluster with kubeadm
26. Get logs for a pod
27. Deploy a pod with the wrong image name (like --image=nginy) and find the error message.
28. Get logs for kubectl
29. Get logs for the scheduler
30. Restart kubelet  
      
    **Non-K8S**
31. Convert a CRT to a PEM  
     a. Convert it back
32. Backup an etcd cluster
33. List the members of an etcd cluster
34. Find the health of etcd