CKAD EXAMINATION ENVIRONMENT TIPS

Tip #1: The process is straightforward. After registering at CNCF and picking a suitable date and time, you will be sent a link to the exam environment that will be provisioned before the exam. You will be also required to install a Chrome plugin with broad permissions, don't forget to delete it afterwards.

The exam itself is proctored, so you'll have someone monitoring your desktop, system processes; the mic and webcam will be turned on for the whole duration. You'll have to show the proctor your ID, the room and the desk and you're required to be alone if you're taking it from home. In my case, all communications with the proctor were made via chat, which is handy in case of a language barrier.

Tip #2: Interruptions. You should expect some, I was asked to 'stop reading questions out loud' and 'move closer to the camera' a couple of times. When I had a glitch with my Internet connection, a simple page refresh fixed it

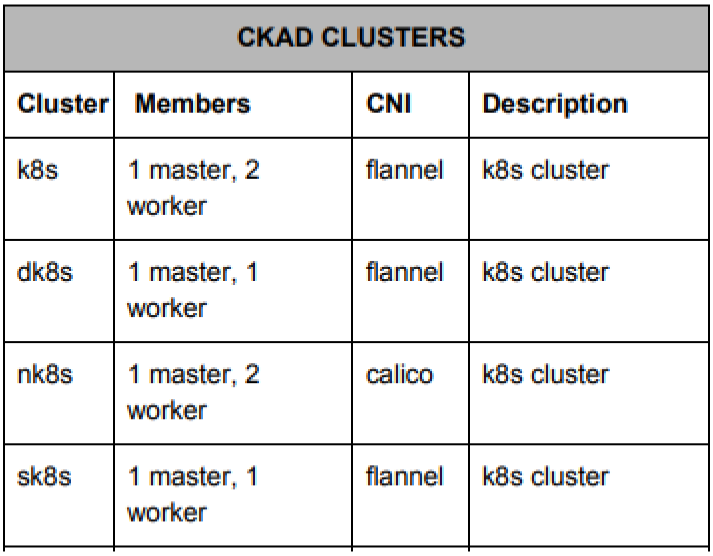
What to Expect:

Two hours and about 19 hands-on problems (6 minutes per task!) that can be tackled in any order, and in any way you prefer as long as it's doable via the in-browser SSH session. You're welcome to write YAML, JSON, run `$ kubectl` , make direct call APIs via `$ curl` , anything, **as long as it solves the problem**.

You will be given several pre-built Kubernetes environments (I believe I had around five), some environments will be empty for you to deploy to, some — packed with stuff for you to modify. Make sure to pay attention to namespaces — each question will require you to use a specific one.

Tip #3. Jumping between clusters. Become familiar with switching contexts and using multiple namespaces. In my case, every task included a fully formed `$ kubectl set-context...` Command, **make sure to run it even if you think you're in the right environment**. Not doing that cost me at least 10 minutes in rework.

There are four clusters (CKAD) that comprise the exam environment, made up of varying numbers of containers, as follows:



CKAD CLUSTERS			
Cluster	**Members**	**CNI**	**Description**
k8s	1 master, 2 worker	flannel	k8s cluster
dk8s	1 master, 1 worker	flannel	k8s cluster
nk8s	1 master, 2 worker	calico	k8s cluster
sk8s	1 master, 1 worker	flannel	k8s cluster

https://training.linuxfoundation.org/wp-content/uploads/2019/08/Important-Tips-CKA-CKAD-Master-8.5.19.pdf

Tip #4. Skim through all tasks first. You're not expected to know everything, be okay with reordering and skipping, each task is independent of others and has a different score. It makes sense to spend a few minutes to skip through all of them and choose the approach that works for you. My mistake was to do the whole thing sequentially, I ended up with a couple of well-paying yet straightforward tasks towards the very end with only a minute left on the clock

Did I mention that everything is browser based? You're allowed two browser tabs: the exam environment itself and the official documentation or github or blog. The environments are connected to the Internet, but you're not allowed to run external scripts hosted outside of these URIs. For example, I have a `$ .bashrc` sitting in my GitHub that sets things up the way I like, you probably have one too. Forget about running it, the proctor won't like it. You'll be dealing with a pretty vanilla environment, except that `$ kubectl` auto completion was set up and the vim indentation settings were OK. Don't expect any $ kubectl bash aliases and if you're not a fan of vim, make sure you're familiar with formed $ export KUBE_EDITOR='nano' .

Tip #5. Bookmarking. Bookmark the URIs to save time

Tip #6. Say no to Google. Get used to finding stuff using built-in search, you won't be allowed to use Google

Tip #7. Study samples. Study and deploy documentation in the docs and on GitHub, there's a lot in there you will reuse during the exam

Exam Environment:

At the time of writing, CNCF uses Gate One to deliver the SSH terminal for CKAD and CKA exams — you're more than welcome to give it a try beforehand. Probably one thing worth noting is that you'll be required to use **[Shift+Insert]** to paste, **[Ctrl+Insert]** to copy and **[Ctrl+Del]** to cut.

At the time of writing, Gate One's demo access was broken, you can try Azure Cloud Shell for a similar SSH experience. Or, you can try Katacoda make Kubernetes environment in your browser.

Tip #8. Keyboard shortcuts. These shortcuts work out of the box in Windows and Linux alongside the more familiar ones; it's a good idea to get used to using them before the exam — you'll be copying and pasting a lot of yaml

The exam environment comes with a handy notepad that I used heavily throughout the exam — it doesn't mess up yaml indentations and overall works flawlessly. In hindsight, I should not have cleared it with every new question and kept appending yaml — there was an opportunity to partially reuse solutions across questions.

Some principles that worked really well for me:

1. **Yaml-first.** When deploying new resources, I would through together a yaml config in the in-browser notepad first, starting either from scratch or reusing samples from Kubernetes docs and then copy and paste it into `$ vim`
2. **No in-place modifications.** When modifying existing resources, I would avoid making in-place edits via `$ kubectl edit…`, instead exporting to yaml via `$ kubectl get … -o yaml` `$ –export` and making edits in `$ vim` or the in-browser notepad
3. **One question — one yaml.** In my home directory I created a yaml per task (01.yaml, 02.yaml, etc.), which proved super helpful when I applied a botched deployment and had to delete the whole thing with a single `$ kubectl delete –f…` command
4. **Dry runs.** It's self-explanatory really, make sure to always do a dry run using `$ –dry-run` before applying your configuration

Tip #9. Bash aliases. If you can't live without bash aliases for `$ kubectl` and its verbs, try to un-learn those at least for the exam; you will be dealing with a pretty vanilla environment and won't be allowed to run an external script to set them up

Tip #10. Speed up deletions. When deleting resources, use `$ –grace-period=0 –force` to speed things up

Preparation:

There's no shortage of materials out there, make sure to stick to the syllabus(for instance, there's no mention of istio yet). CNCF also offers two paid courses, Kubernetes for Developers and Kubernetes Fundamentals that are aligned with the CKAD and CKA exams respectively. Nothing beats real-life practice though, ideally you should be able to score at least 50–55% without any preparation whatsoever.

Tip #11. Know the syllabus, don't overlearn. Be familiar with the exam syllabus and study the Candidate Handbook

Tip #12. Forget about the Kubernetes Dashboard. Don't become attached to the Kubernetes Dashboard, you won't see it used often in real life and you definitely won't be using it during the exam

Tip #13. Practice. Get some hands-on exposure before attempting the exam to benefit from 'muscle memory' and habits. If you don't use Kubernetes at work, come up with a personal project and work on it for a while

Tip #14. Free retake. Keep in mind that all exams purchased directly from CNCF, come with one retake for free

Good luck, thanks for reading and until next time!

SOURCE:

https://medium.com/@ameenalam202/tips-for-successful-certified-kubernetes-application-developer-ckad-exam-preparation-efaed88500fe

https://medium.com/@iizotov/exam-notes-ckad-c1c4f9fb9e73

Note: It is all of my research about the topic. I am not a content writer, a copy is short and persuasive.