I faced the Certified Kubernetes Administrator (CKA) exam on 26th May 2025. The CKA exam was changed after February 2025. I got to know about this around two weeks before my exam date — that was my mistake, actually, for not exploring more and just following the course content on Udemy. (I’m writing this within the first 24 hours after my first attempt, and I still haven’t got my results.)

February 2025 Changes — [Certified Kubernetes Administrator (CKA) Program Changes — Linux Foundation — Education](https://training.linuxfoundation.org/certified-kubernetes-administrator-cka-program-changes/)

I completed the CKAD in November 2023 and shared my experience in <https://medium.com/@dilancs/my-ckad-story-4a10da7de1b9>.

The content there may also be useful, but compared to that, my CKA experience is a little different. So, I’ll begin from how I started.

**Main References**

[Certified Kubernetes Administrator (CKA) Practice Exam Tests | Udemy](https://www.udemy.com/course/certified-kubernetes-administrator-with-practice-tests/)

I started with the above course on Udemy. As I was facing a kubernetes after 1.5 years, I had lost most of my CKAD knowledge from memory. So I had to refresh, and I followed the whole course from the beginning. The Udemy course provides access to the following lab series, which is a must-do.

[CKA Certification Course — Certified Kubernetes Administrator Course | KodeKloud](https://learn.kodekloud.com/courses/cka-certification-course-certified-kubernetes-administrator)

**Must Follow YouTube Resources**

[2025 CKA Exam Questions & Solutions UPDATE! | Full Walkthrough!](https://www.youtube.com/watch?v=eGv6iPWQKyo) [JayDemy — YouTube](https://www.youtube.com/@jaydemytech/videos)

**Exam Question Preparation**

I started with the one Lightning Lab session available in the above KodeKloud Labs and then did the 3 Mock Exams there as well. In the first attempt, these can also be hard to complete within the time limit, but you can retry them and should reach a confident level where you can do them easily within the time. Repeating is not a problem. More practice will make you more comfortable.  
Then, the CKA exam registration gives you two exam simulator sessions, each valid for 36 hours. **These simulator sessions are a must**.

[**Killer Shell — Exam Simulators**](https://killer.sh/cka)

The simulator sessions are different from the CKAD simulator sessions. CKAD gives you the same set of questions in each simulator session, but CKA gives you a different set of questions in each simulator. I recommend utilizing both before your first attempt — one about a week before, and the second one just before the exam (the 36-hour window ends on the exam day).

I made a mistake thinking these sessions were similar to CKAD and would repeat the same questions. I took the first one two days before the exam, keeping the other in case I needed a retake, assuming both sessions would have the same set of questions. I found out just two days before the exam that the second session has a different set of questions, so I had to do both in parallel. So, do make sure to use both sessions before the exam — they cover different scenarios.

Each session gives you 16 or 17 questions. In the first attempt of each, I could only solve about half of the questions. Don’t panic — that’s totally fine. Study the questions and answers, then repeat the sessions. Do them at least three times or more within the 36 hours. I could only do three attempts per session, as I had to do both in parallel because of my mistake. Try to reach a point where you can complete all the questions within two hours. My best time was 1 hour and 45 minutes.

**My next mistake**  
I googled about the exam around 4–5 days before taking the two simulator sessions and found that the exam had changed after 2025. The theory content was covered in the course, but not all of it was in the Mock Exams. So, very luckily, I found the following YouTube video session:

[2025 CKA Exam Questions & Solutions UPDATE! | Full Walkthrough!](https://www.youtube.com/watch?v=eGv6iPWQKyo)

The above video gave me a totally different perspective regarding the questions, and a comment on the video made me explore more. That’s how I found the following series as well:

[JayDemy — YouTube](https://www.youtube.com/@jaydemytech/videos)

As people commented in the videos, the exact same types of questions appeared in the exam. So, the above two video sessions are a must-do.

**Actual Exam Environment Experience**

The Actual Exam Remote Desktop is almost similar to the two killersh simulator sessions given which I have mentioned above. So here few things to note.

1. In my CKAD exam, for each question, I had to switch the context while staying in the same terminal session. So when I started the exam, I saved my aliases into the ~/.bashrc to reuse those shortcuts throughout the exam, which saved a lot of time. But in CKA, it is instructed to SSH into a separate node in each question from a base terminal session given to you. So trying to save the aliases into .bashrc 16 times for all questions will kill your time. Typing the full commands will be better. What I did was open a test file using MousePad and wrote all the alias shortcuts there when the exam began. Then, after SSH-ing into the cluster node given in each question, I copied and pasted the alias statements directly into the terminal without trying to save them in .bashrc.
2. You can copy and paste in the exam’s Linux terminal using CTRL+SHIFT+C and CTRL+SHIFT+V or using right-click menus. But we are all totally used to **Copy on Select** and **Paste on Right Click**. These two options were not enabled in the terminal by default. Using the right-click menus will kill your time. In the terminal preferences, you can enable both **Copy on Select** and **Paste on Right Click**. This was available in the Killer.sh simulator sessions, but I was not sure about the actual exam. So I practiced CTRL+SHIFT+C and CTRL+SHIFT+V shortcuts in my last simulator session before the exam. But in the exam, luckily, changing the terminal preferences is allowed. That was a total relief for me, because I’ve been used to **Copy on Select** and **Paste on Right Click** for years.
3. The question section allows you to copy certain keywords from there to paste into the terminal. I noticed a delay of around 1–2 seconds before they are copied to the clipboard. If you click on the question keyword and immediately paste it into the terminal, it may not paste correctly — the previously copied text gets pasted instead. If you delete that and paste again, then the latest copied text gets pasted. It took me some time to exactly figure out this behavior. So, to handle this efficiently, I followed a trick: I clicked on the contents in the question panel while keeping something ready to be typed manually in the terminal, so I’d automatically get that delay between copy and paste. When that wasn’t possible, I waited 1–2 seconds before pasting. This helped me manage my time better.
4. The resolution was lower than in the Killer.sh simulator, so I could see less content on the screen. Also, when using the provided Firefox browser to view the documentation, scrolling using the mouse wheel was not smooth.
5. The Firefox browser opens with the following URL:  
   [Resources Allowed: All LF Certification Programs | T&C DOCS (Candidate Facing Resources)](https://docs.linuxfoundation.org/tc-docs/certification/certification-resources-allowed)  
   I also had a doubt about how to remember the URLs for Helm and Gateway API, but these were provided in that page.  
   **Note**: The documentation for Kustomize ([The Kustomization File | SIG CLI](https://kubectl.docs.kubernetes.io/references/kustomize/kustomization/)) is not allowed in the exam. In Mumshad’s Udemy course, some content relies on that documentation, but in the exam, only this is allowed:  
   [Declarative Management of Kubernetes Objects Using Kustomize | Kubernetes](https://kubernetes.io/docs/tasks/manage-kubernetes-objects/kustomization/)  
   So I think the questions will not include syntax that’s only found in the Udemy course. I unnecessarily tried to remember them, but since they can’t be used in the exam, there’s no point in memorizing them.
6. Make sure you have enough disk space on your machine. I got a Windows “low disk space” notification during the exam. I ignored the message and continued. After the exam, only around 5 GB of space was available. So I suggest keeping 15–20 GB of free disk space, as I now know how much space the PSI software takes.  
   Also, make sure your broadband router and computer are connected to a UPS. It’s even better if you can keep a mobile 4G dongle connected as a backup, in case your broadband fails in the middle — then the connection can switch to mobile data.

**The questions I Got**

I got 16 questions, and all of these are based on some scenario which was already built in each question’s cluster.

1. Select one out of the three NetworkPolicy manifests which matches the scenario described in the question (e.g., backend pods should only have ingress traffic from the frontend namespace).
2. Install the cridocker .deb package using the dpkg -i command and perform some follow-up tasks.
3. List all CRDs matching a keyword (cert-manager) and write them into a file. Then, document a field from the spec of the CRD using the kubectl explain command.
4. Expose a deployment using a NodePort service.
5. Create an HPA (Horizontal Pod Autoscaler) for an existing deployment.
6. Create an Ingress resource matching the scenario described in the question.
7. Create a Gateway with TLS and an HTTPRoute, matching the existing Ingress resource in the environment. Delete the Ingress after creating the Gateway.
8. Generate a Helm template and save it to a file using the helm template command. Then install the Helm chart with some changes to the Helm values. Both tasks were in a specific namespace, and a specific chart version was mentioned.
9. Create a PriorityClass with a modification compared to an existing user-defined PriorityClass.
10. Create a StorageClass.
11. Create a PVC and attach it to a Pod. There was an existing PV in the environment, and we had to choose PVC properties to match the PV.
12. Add a sidecar log container to an existing deployment by mounting a shared volume.
13. Change the ConfigMap properties of an existing NGINX ConfigMap to enable both TLSv1.2 and TLSv1.3. Only TLSv1.3 was enabled initially.
14. A deployment with three replicas had some pods in a pending state because the resource requests of containers exceeded the resources available on the node. Check the node’s CPU and memory, then divide them equally among the containers — keeping some overhead for system components and buffer — so that the deployment schedules all three replicas without any being pending.
15. kube-apiserver and kube-scheduler in a cluster were not working, but etcd, kube-controller-manager, and kubelet were. Troubleshoot and fix the issue.
16. Choose a CNI between Flannel and Calico that has built-in support for Network Policies (Calico supports them). Install the CNI and configure it to work with the current node’s PodCIDR.

The order of the questions is not the the order i got questions in the exam.

Compared to CKAD, I have to say that answering these questions in the given time is tough. With my whole CKAD and CKA experience, even when we are 100% sure about the answers, there will always be mistakes that we get wrong. This happened in all my CKAD and CKA mock exams and Killer.sh simulator sessions. Even in my CKAD exam, I completed it in 1 and a half hours and found three mistakes during my 30-minute review and corrected them (please refer to the end part of [*My CKAD Story*](https://medium.com/@dilancs/my-ckad-story-4a10da7de1b9)). Still, I got 88, but I was thinking I had answered everything correctly.

In my CKA, I failed to answer the last three questions — 14, 15, and 16 — and I didn’t have any time left to review the ones I had already answered. Maybe that’s because I spent too much time troubleshooting those last three. So, I highly recommend trying to answer all questions while keeping at least 10–15 minutes at the end to do a final review of the answered ones.

**The Two Most Common Mistakes**

1. **Forgetting to SSH into the mentioned cluster** — Because we have to manually exit from the previous question’s cluster, go back to the base terminal, and SSH into the new cluster for each question. I made this mistake once: I tried to create the PriorityClass in the previous cluster, but the priority value had to be taken from an existing user-defined PriorityClass. There were no user-defined PriorityClasses in the previous cluster—only system ones. So, I tried to use a system PriorityClass value, but kubectl gave an error. While being stuck there, I realized the mistake. I have no idea whether I did the same in any other question since I had no time for a full review.
2. **Forgetting to switch to the namespace mentioned in the question** — It’s time-consuming to type -n <namespace> in each command, so I switch to the namespace using kubectl config set-context, and I use an alias for that:  
   alias kns='kubectl config set-context --current --namespace'

**The Questions I Could Not Answer from the Above List**

1. (No 14 of above) — The deployment manifest was too bulky because it had unnecessary length due to extra new lines. It also included Init Containers. I tried to divide the resources properly, but something went wrong every time, so the pods didn’t become ready. Probably my mistake, but I’m sharing it anyway.
2. (No 15 of above) — In the kube-apiserver.yaml, the etcd-server URL was wrong—I corrected it. That was the only issue I found. All certificate paths were correct. But still, the kube-apiserver did not start, and I couldn't find the root cause of the kube-scheduler failure either.
3. (No 16 of above) — The Calico tigera-operator.yaml URL was provided. The custom-resources.yaml URL was not, but we needed it to configure the PodCIDR in the Installation CRD. Anyway, I remembered the path for custom-resources.yaml as it was in the same directory. But even then, errors were thrown when I did kubectl apply with the tigera-operator.yaml. The following error came:

“installations.operator.tigera.io” is invalid: metadata.annotations: Too long:

*I didn’t see any problem with the annotations of the installations.operator.tigera.io CRD either, but I still tried to remove the annotations and install it again—it still failed. I reported a ticket regarding this issue as well. I will post the response once I get it. So, knowing the custom-resources.yaml URL didn’t help, as the initial installation failed anyway.*

Because of the above three issues, I wasted a certain amount of time, so I missed the chance to review my other answers. It would have been better if I had just skipped these three — then I might have been able to correct any other mistakes in the 13 questions I answered.