



Jak zostać Kubestronaut'ą...

1. Wstęp

Cześć, nazywam się Konrad Nowacki i zostałem Kubestronautą, to znaczy zdałem 5 certyfikatów Kubernetes organizowanych przez Linux Foundation. Co istotne, wszystkie certyfikaty są ważne, w momencie pisania tej instrukcji każdy z nich jest wydawany na 2 lata.

W dalszej części będę używał skrótów **CK*** aby odnieść się do certyfikatów **CKAD**, **CKA** oraz **CKS**, czyli do tych trudniejszych egzaminów oraz **KC*** w odniesieniu do **KCNA** i **KCSA**, czyli tych łatwiejszych.

Będę też używał zamiennie terminów **certyfikat** i **egzamin**.

2. O egzaminach

KC* to egzaminy teoretyczne w formie odpowiedzi pojedynczego wyboru. Można do nich podejść 2 razy w ramach jednego zakupionego vouchera. Wyniki dostępne są w ciągu kilku minut od zakończenia.

CK* to egzaminy praktyczne, w formie Linuxa, gdzie wszystkie zadania muszą być zrobione w terminalu. Do dyspozycji mamy także przeglądarkę z Firefox z oficjalną dokumentacją Kubernetes. Wyniki są dostępne do 24h po zakończeniu, w moim przypadku nigdy nie dostałem ich szybciej niż po 22h.

Podczas każdego egzaminu CK* pojawiły się zadania z Network Policy i Ingress

KCNA - najłatwiejszy egzamin, warto od niego zacząć jeśli ktoś ma niewiele wiedzy o Cloud Native oraz Kubernetes. Poniżej podaję kurs Jamesa Spurina, który bardzo polecam podczas przygotowania pod KCNA.

KCSA - egzamin podobny do KCSA, jednak skupia się na aspektach bezpieczeństwa. Główna trudność polega na braku dedykowanych materiałów, żeby się przygotować. Niektóre pytania były całkiem szczegółowe, inne z kolei banalne.

CKAD - pierwszy egzamin praktyczny. W sylabusie wyczytamy, że trzeba znać Kubernetes i Docker i faktycznie, było zadanie z Dockera (export image do pliku). Pytania oscylują wokół tworzenia, zarządzania i usuwania resource'ów kubernetesowych. Warto znać imperatywne komendy terminalowe celem szybkiego tworzenia resource'ów, często wszystko co trzeba zrobić da się zawrzeć w takiej komendzie. A w miarę potrzeby i tak można z niej wyciągnąć manifest .yaml (`--dry-run=client -oyaml`).

CKA - Wg mnie poziom trudności bardzo podobny do CKAD, jednak community np na Reddit często porównuje CKA do CKS pod tym względem. Chociaż sylabus wspomina o znajomości Dockera, Helma i Istio, 100% pytań które mi się trafiły dotyczyły czystego Kubernetesa. Jest kilka zadań, które z bardzo dużym prawdopodobieństwem się pojawią i aby je rozwiązać, wystarczy dobrze przejść opis z dokumentacji. Oto one:

Upgrade node

<https://kubernetes.io/docs/tasks/administer-cluster/kubeadm/kubeadm-upgrade/>

ETCD encryption

<https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/>

Kube-api server options

<https://kubernetes.io/docs/reference/command-line-tools-reference/kube-apiserver/>

CKS - zdecydowanie najtrudniejszy egzamin. Jedną z trudności jest zapoznanie się z wieloma narzędziami wspomagającymi K8s np. Falco, app_armor, bom. Sporo pytań dotyczyło jakiejś konfiguracji Audit Policy i Audit Logging, szyfrowania, pojawił się też temat upgrade node taki sam jak na CKA. Pojawił się też temat czysto Linuxowym - trzeba było dodać user do grupy.

Audit logging & policy

<https://kubernetes.io/docs/tasks/debug/debug-cluster/audit/>

Ingress TLS http/https redirect

<https://kubernetes.io/docs/concepts/services-networking/ingress/>

Wiele pytań było mniej lub bardziej związanych z ServiceAccount...

<https://kubernetes.io/docs/concepts/security/service-accounts/>

... oraz RBAC

<https://kubernetes.io/docs/reference/access-authn-authz/rbac/>

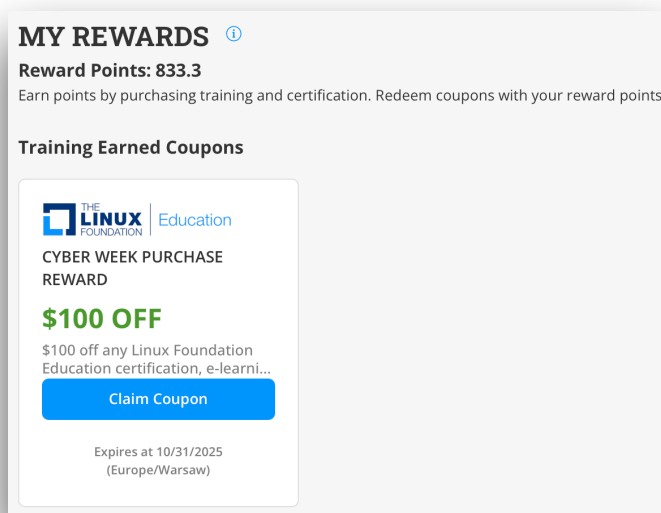
3. Ceny

Certyfikaty Linux Foundation są całkiem drogie, dlatego warto poznać kilka sposobów, aby obniżyć ich cenę nawet o 50%.

Przede wszystkim social media i newsletter e-mail - kilka razy w roku można znaleźć zniżkę **35-50%**, z czego od kilku lat w okolicach **Black Week** i **Cyber Week** jest zawsze duża zniżka 50% na większość produktów.

Kolejną opcją są zniżki otrzymane po poprzednich zakupach. Po odpowiedniej ilości \$\$ wydanych na certyfikaty. I tutaj dobra wiadomość, ale pewien nie jestem: ilość wydanych \$\$ która jest wzięta pod uwagę podczas zakupów to oryginalna cena, bez zniżek. Czyli jak na Black Friday kupimy KC* za \$125 to dostaniemy \$250 w punktach do naszego konta.

Po uzyskaniu tytułu Kubestronauty dostałem 5 voucherów na 50% ważnych do końca 2025, chętnie się podzielę :)



MY REWARDS ⓘ

Reward Points: 833.3

Earn points by purchasing training and certification. Redeem coupons with your reward points.

Training Earned Coupons

THE LINUX FOUNDATION | Education

CYBER WEEK PURCHASE REWARD

\$100 OFF

\$100 off any Linux Foundation Education certification, e-learn...

[Claim Coupon](#)

Expires at 10/31/2025 (Europe/Warsaw)

4. Kolejność egzaminów

Wg poziomu trudności, zalecana kolejność podchodzenia do egzaminów to

KCNA -> KCSA -> CKAD -> CKA -> CKS

Jedynym wymogiem formalnym jest CKA -> CKS - trzeba mieć ważny zdany CKA aby móc umówić się na CKS.

Ja podszedłem w innej kolejności:

CKAD -> CKA -> CKS -> KCSA -> KCNA

Czyli najpierw trudniejsze a na deser KC* czyli łatwe. Tak też było, KC* zrobiłem w ciągu tygodnia mając 85% wiedzy po nauce do CK*

5. Materiały do nauki

Killer.sh (<https://killer.sh/>) - najlepszy symulator dla CK*, mamy dostęp do panelu bardzo podobnego do tego z rzeczywistego egzaminu. Pytania są trudniejsze i jest ich więcej niż na oficjalnym podejściu, więc jeśli umiesz zdać te na killer.sh to bardzo dobry wskaźnik, że możesz podejść do oficjalnego. Zła wiadomość, że dostęp do egzaminu i odpowiedzi jest tylko przez 36h, dlatego polecam sobie skopiować (ja sobie wydrukowałem, żeby móc nabazgrolić notatki długopisem), aby móc się z nich uczyć dłużej. Dobra wiadomość natomiast - nie trzeba tu nic kupować. 2 podejścia dostajemy kupując CK*. Inna dobra wiadomość, jeżeli chcielibyśmy dokupić sobie jeszcze 1 podejście, cena jest niższa (\$10) niż oryginalna za pierwszy zakup.

- Autor Killer.sh jest też autorem tego kursu do CKS: (<https://www.youtube.com/watch?v=d9xfB5qaOfg>)

Gdzieś przeczytałem, że kurs jest out of date i nie uwzględnia najświeższych wymagań, ale i tak nic lepszego nie znalazłem do nauki CKS.

- Podczas nauki CKS obejrzałem też tę playlistę:

https://www.youtube.com/watch?v=x_klpjAilMs&list=PL5uLNcv9SibBu_KgCL-OSHzFFzlafKm6F

- Oraz tę z rozwiązaniami z killer.sh CKS:

https://www.youtube.com/watch?v=Jd_j2wrusz6E&list=PLpbwBK0ptssx38770vYNwZEUceGNw54CH

https://www.youtube.com/watch?v=Jd_j2wrusz6E&list=PLpbwBK0ptssx38770vYNwZEUceGNw54CH

Killercoda (<https://killercoda.com/>) - playground dla Kubernetesa a także dla każdego z CK*. Darmowy, z pewnymi ograniczeniami czasowymi. Kandydat do CK* powinien bardzo sprawnie rozwiązywać i rozumieć większość scenariuszy, które się tam znajdują.

KodeKloud:

- <https://www.udemy.com/user/mumshad-mannambeth/>

- <https://kodekloud.com/>

Firma KodeKloud ma bardzo dobre kursy do CKAD i CKA dostępne na Udemy, więc można na firmowym koncie odpalić, CKS natomiast jest dostępny tylko przez ich stronę. Bardzo dobre wprowadzenie teoretyczne i kilka wartościowych zadań praktycznych.

James Spurin (<https://www.udemy.com/user/james-spurin/>) - autor zdecydowanie najlepszego kursu pod KCNA. Ogrom testów próbnych, bardzo zbliżonych do faktycznego egzaminu.

Reddit - niedocenione przeze mnie źródło informacji na początku mojej drogi. Bardzo polecam czytać wszystkie wątki dot. egzaminu, do którego będziecie podchodzić. Czasami ktoś wspomni

o zadaniu które miał albo o technice, jak je rozwiązał. Linux Foundation ma ograniczoną liczbę zadań, więc możliwe, że się trafi na coś podobnego.

Podsumowując:

KCNA - James Spurin

KCSA - Tutaj znalazłem najmniej materiałów do nauki, przerobiłem te pytania <https://www.udemy.com/course/kcsa-kubernetes-cloud-native-security-associate-exam-prep/?couponCode=MT40825> i wystarczyło.

CKAD - KodeKloud na Udemy, Killer.sh, Killercoda

CKA - KodeKloud na Udemy, Killer.sh, Killercoda

CKS - Kurs autora Killer.sh, Killer.sh, Killercoda, rozwiązania killer.sh na Youtube

6. Inne uwagi

- **notatki po egzaminie** - polecam po każdym egzaminie od razu usiąść do notatnika, ja robiłem to w formie papierowej, i spisać wszystko co pamiętamy po egzaminie, szczególnie to czego nie umieliśmy zrobić. W przypadku niezdania, mamy z czego się przygotować. Jest szansa, że na drugim podejściu jakieś pytanie, a przynajmniej typ zadania się powtórzy.
- **Czas** - czyli to czego na CK* brakuje najbardziej. Jeśli mamy 120 minut na 16 zadań, to daje 7,5 minuty na zadanie. Uwzględniając, że czasami potrzeba chwilę czegoś poszukać w dokumentacji, która laguje, można łatwo przepalić 15 minut na jeden krok. A brak czasu spowoduje większy stres pod koniec, co może wpłynąć na całość. Polecam oflagować takie zadanie i wrócić do niego pod koniec.
- **Dokumentacja** - podczas egzaminów CK* można korzystać z oficjalnej dokumentacji Kubernetes (<https://kubernetes.io/docs/home/>) (Podczas CKS także Falco). Warto podczas treningu zapamiętać sobie, jakie hasło wpisać w tę dokumentację aby szybko zlokalizować to, co chcemy skopiować. Warto też zwrócić uwagę na indentację, żeby potem nie marnować czasu na formatowanie w VIMie.

Oczywiście podstawową skarbnicą komend jest cheatsheet, przydatne do początkowych komend oraz to wielu zadań na CKAD i CKA

<https://kubernetes.io/docs/reference/kubectl/quick-reference/>

- **Kopiowanie** - coś pozornie łatwego, jednak może przepalić sporo czasu jak się i tego nie przygotuje. Ja egzaminy pisałem na MacBooku, więc byłem przyzwyczajony do kopiowania i wklejania CMD-C/CMD-V. Ale podczas egzaminów CK* mamy Linuxa i terminal, więc w terminalu używamy CTRL-SHIFT-C/CTRL-SHIFT-V. Ale to nie koniec niespodzianek, bo w Firefoxie kopiujemy CTRL-C/CTRL-V. Czyli kopiując z dokumentacji do VIMa musimy użyć 2 różnych kombinacji klawiszy. CTRL-C oraz CTRL-SHIFT-V
- **Vim, terminal** - Kubernetes oczywiście jest najważniejszą, ale nie jedyną technologią potrzebną do zdania egzaminu CK*. Jako, że całość odbywa się w terminalu, warto być zaznajomionym z podstawami jego obsługi. Komendy **cd, sudo -i, grep, touch, >, >>, cat, echo, ls, ps, cp, mv** to podstawa sprawnego poruszania się po terminalu. Dużą częścią egzaminu stanowi przygotowywanie manifestów w formacie yaml, do czego potrzeba edytora tekstu. W terminalu jest VIM i Nano, ja używałem VIMa, bo z Nano szło wszystko zbyt wolno, nie było kolorowania składni i numerków po boku. Ogólnie VIM jest trudniejszy w nauce, ale jak już się ogarnie podstawy to dużo szybciej idzie obsługa. A czas jest cenny :wq!
- **Lagi i rozdzielczość Firefoxa** - Podczas CK* można korzystać z oficjalnej dokumentacji Kubernetes, która jest dostępna w Firefox'ie w Linuxie na którym rozwiązujemy zadania. Jednak nie jest to bardzo przyjemne doświadczenie, są lagi, nie wszystko działa płynnie. Warto też zwrócić uwagę na rozdzielczość Firefoxa. W moim przypadku okno Firefox'a otworzyło się na 110% wysokości dostępnego ekranu, tak, że po wciśnięciu CTRL-F celem wyszukania tekstu, panel do wpisania tekstu był w dolnej części strony, która była niewidoczna.

- **Killer.sh print** - symulator egzaminu oraz rozwiązania są dostępne przez 36h. A takich podejść mamy tylko 2, dlatego warto zachować sobie te rozwiązania żeby móc do nich wrócić po tych 36h. Ja zrobiłem save web as PDF i później analizowałem 'na sucho'.
- **YAML** - jako, że manifesty piszemy albo generujemy w formacie yaml warto upewnić się, że dobrze rozumiemy tę strukturę.
- **Początkowe komendy** - aby maksymalnie ułatwić sobie start egzaminu, warto umieć szybko stworzyć potrzebne aliasy. Ja używałem takich:
 - alias kg="kubectl get"
 - alias kdp="kubectl describe pod"
 - alias kn="kubectl config set-context --current --namespace " (przykład: kn my-namespace)
 - export do="—dry-run=client -oyaml" (przykład: k run mypod —image nginx \$do > file.yaml)
- **2 terminale** - w CK* nie chodzi o zapamiętywanie, tylko o umiejętność znalezienia tego, czego potrzebujemy. Podstawową wiedzą jest dostęp do dokumentacji. Podczas egzaminu korzystałem także z komendy **kubectl explain <resource>**. Miałem ją otwartą w drugim oknie terminala, żeby nie zaciemniać sobie np. otwartego pliku yaml, który muszę zmodyfikować. **kubectl explain <resource>** jest szczególnie przydatny z opcją —recursive. Więcej tutaj (https://kubernetes.io/docs/reference/kubectl/generated/kubectl_explain/)
- **Namespace oraz context** - niesamowicie ważna sprawa, bo łatwo o niej zapomnieć. Podczas egzaminów CK*, każde zadanie musi być wykonane w odpowiednim kontekście oraz namespace. Kontekst jest zawsze podane w czerwonej ramce na początku treści zadania, namespace natomiast w treści zadania. Dlatego warto mieć taki **alias kn**, aby efektywnie móc się przełączać. **Wykonanie zadania w złym namespace albo context da 0 punktów za zadanie.**
- **Grep** - jedno z bardziej przydatnych narzędzi w terminalu. Czy to korzystanie z dokumentacji man, kubectl ... —help, kubectl explain czy cat, warto znać, zwłaszcza z różnymi flagami. -A, -B, -C, -e przydały mi się podczas każdego CK*.