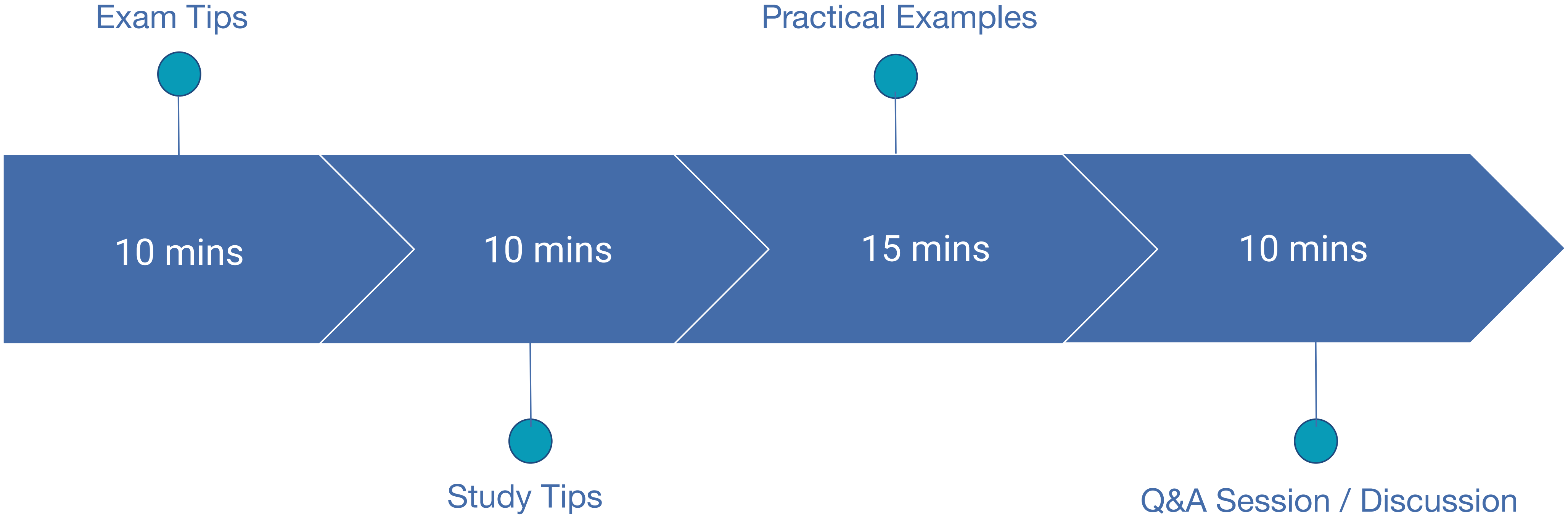
CLOUD NATIVE
COMPUTING
FOUNDATION

CKAD Study & Exam Tips

https://community.cncf.io/manly

Agenda


Exam Tips
10 mins
Study Tips
10 mins
Practical Examples
15 mins
Q&A Session / Discussion
10 mins

Special thanks

SIG Docs

https://kubernetes.io/docs/contribute/participate

slack.cncf.io

slack.k8s.io

Exam Tips - Why sit the exam

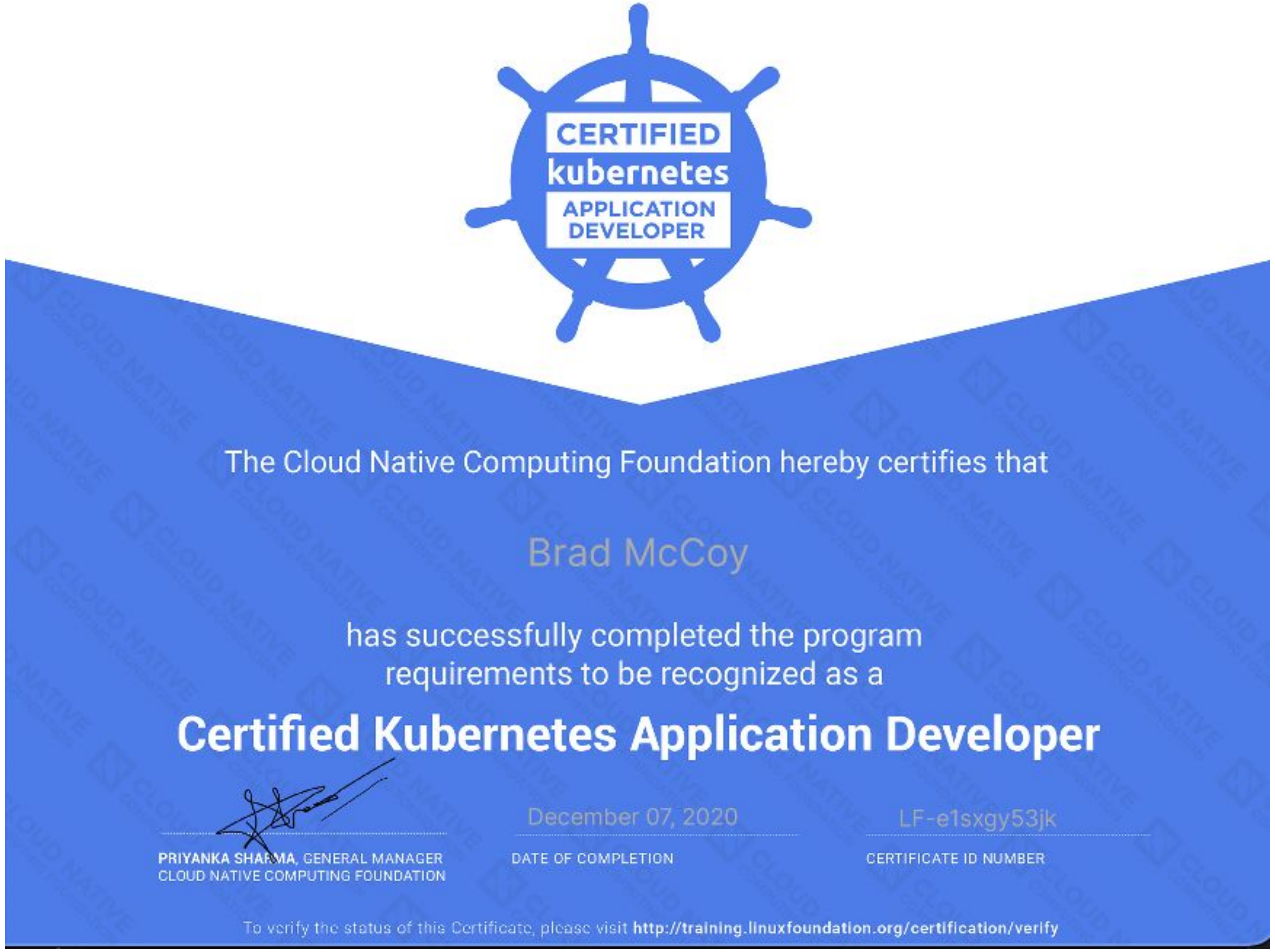
As one of the highest velocity projects in the history of open source, Kubernetes use is exploding. The Cloud Native Computing Foundation is committed to growing the community of Kubernetes-knowledgeable application developers, thereby enabling continued growth across the broad set of organizations using the technology.

Certification is a key step in that process, allowing certified application developers to quickly establish their credibility and value in the job market, and also allowing companies to more quickly hire high-quality teams to support their growth.

Exam Tips - Exam Curriculum

- 13% – Core Concepts
- 18% – Configuration
- 10% – Multi-Container Pods
- 18% – Observability
- 20% – Pod Design
- 13% – Services & Networking
- 8% – State Persistence

CKAD Certificate



CERTIFIED kubernetes APPLICATION DEVELOPER

The Cloud Native Computing Foundation hereby certifies that

Brad McCoy

has successfully completed the program
requirements to be recognized as a

Certified Kubernetes Application Developer

	December 07, 2020	LF-e1sxgy53jk
PRIYANKA SHARMA, GENERAL MANAGER CLOUD NATIVE COMPUTING FOUNDATION	DATE OF COMPLETION	CERTIFICATE ID NUMBER

To verify the status of this Certificate, please visit **http://training.linuxfoundation.org/certification/verify**

Exam Tips

Exam Tips - Booking your exam

To book goto: https://www.cncf.io/certification/ckad

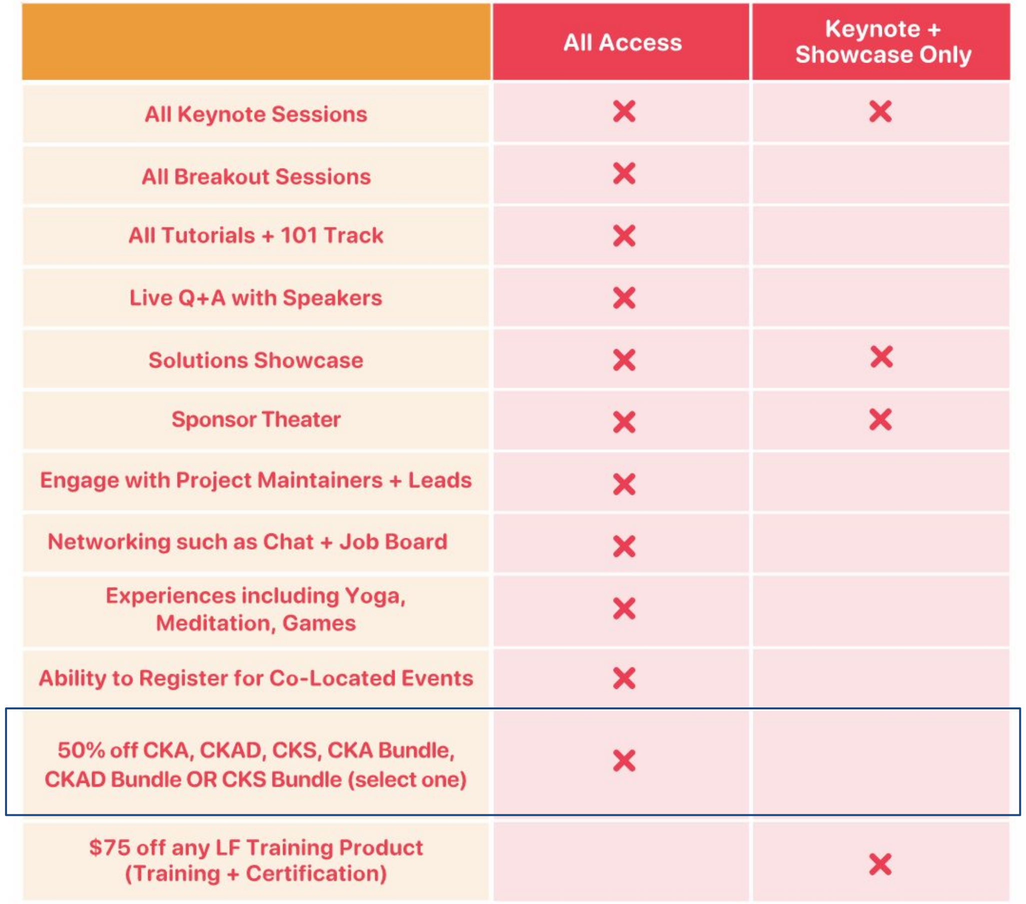
- Book exam straight away to set your goal
- Free reschedule as many times as you want 24 hours in advance
- Price is in US dollars
- One free retake

$300

Exam only

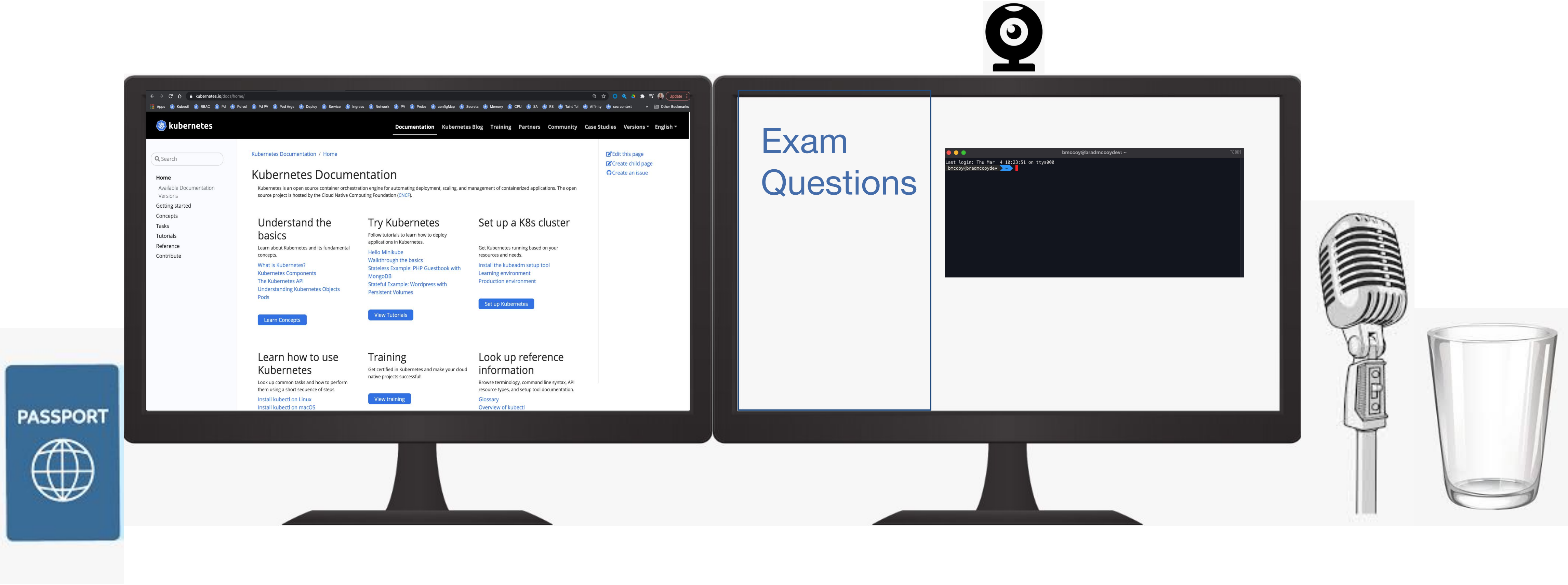
Exam Tips - Discount Codes

https://events.linuxfoundation.org/kubecon-cloudnativecon-europe/register



	All Access	Keynote + Showcase Only
All Keynote Sessions	✕	✕
All Breakout Sessions	✕	
All Tutorials + 101 Track	✕	
Live Q+A with Speakers	✕	
Solutions Showcase	✕	✕
Sponsor Theater	✕	✕
Engage with Project Maintainers + Leads	✕	
Networking such as Chat + Job Board	✕	
Experiences including Yoga, Meditation, Games	✕	
Ability to Register for Co-Located Events	✕	
50% off CKA, CKAD, CKS, CKA Bundle, CKAD Bundle OR CKS Bundle (select one)	✕	
$75 off any LF Training Product (Training + Certification)		✕

Exam Day



Kubernetes Documentation / Home

Kubernetes Documentation

Kubernetes is an open source container orchestration engine for automating deployment, scaling, and management of containerized applications. The open source project is hosted by the Cloud Native Computing Foundation (CNCF).

Understand the basics

Learn about Kubernetes and its fundamental concepts.

What is Kubernetes?
Kubernetes Components
The Kubernetes API
Understanding Kubernetes Objects
Pods

Learn Concepts

Try Kubernetes

Follow tutorials to learn how to deploy applications in Kubernetes.

Hello Minikube
Walkthrough the basics
Stateless Example: PHP Guestbook with MongoDB
Stateful Example: Wordpress with Persistent Volumes

View Tutorials

Set up a K8s cluster

Get Kubernetes running based on your resources and needs.

Install the kubeadm setup tool
Learning environment
Production environment

Set up Kubernetes

Learn how to use Kubernetes

Look up common tasks and how to perform them using a short sequence of steps.

Install kubectl on Linux
Install kubectl on macOS

Training

Get certified in Kubernetes and make your cloud native projects successful!

View training

Look up reference information

Browse terminology, command line syntax, API resource types, and setup tool documentation.

Glossary
Overview of kubectl

```
Last login: Thu Mar  4 10:23:51 on ttys000
 bmccoy@bradmccoydev  ~ 
```

https://community.cncf.io/manly

Exam Tips

- You can start the exam 15 minutes before the scheduled time, this is the amount of time it will take to talk to the proctor.
- Make sure that your room is clean on exam day, they will ask you to walk around the room with your camera
- Nothing to be on the walls
- Drinks need to be in a clear glass, toilet breaks allowed.
- Don’t talk during the exam
- Take note of how to copy and paste

Exam Tips

- There is a notepad you can use in the portal
- The exam consists of 15-20 performance based tasks
- Each question has a weight eg 5% skip hard ones and flag them for later
- Straight away use **sudo -i** to assume elevated privileges to make things easier. If you want to get out type exit or

  su student
- Use “**ssh <Node Name>**” to get into the node, remember which node you are in so you don't get confused
- Always copy pod names etc instead of typing out to avoid spelling mistakes
- Use imperative commands over YAML

Exam Tips - Naming YAML files

When generating YAML files, name them with the question number

Example

1.yaml
1-svc.yaml

```
kubectl run example bradmccoydev/test:latest -o yaml > 1.yaml
```

```
cp 1.yaml 1a.yaml
```
(backup if required)

```
kubectl create -f 1.yaml
```

Exam Tips - alias k=kubectl

alias k=kubectl

Use this so you don't have to keep typing kubectl all the time

Eg.

```
alias k=kubectl
k run nginx --image=nginx
```

Exam Tips

Make sure you are in the right cluster, and the right namespace as specified in the question, to avoid deploying to the wrong namespace when using YAML to create resources specify it in the metadata in case you forget.

Exam Tips - kubectl api-resources

kubectl api-resources to get names/short names *eg. kubectl get svc*

Exam Tips - Reverse Search (Ctrl + r)

Exam Tips - Watch command

Make use of the **watch** command when checking if deployments worked etc it will execute the command every 2 seconds until you **ctrl + c** *eg watch kubectl get pods*

Exam Tips - kubectl -h command

Use the **-h** for help! **kubectl run -h** gives some hints if you forgot imperative commands under pressure.

Practice the **-h** command on other options such as **kubectl create deploy -h** etc

Study Tips

Study Tips - Courses

THE LINUX FOUNDATION

walidshaari/Kubernetes-Certified-Administrator.git

Udemy

KODEKLOUD

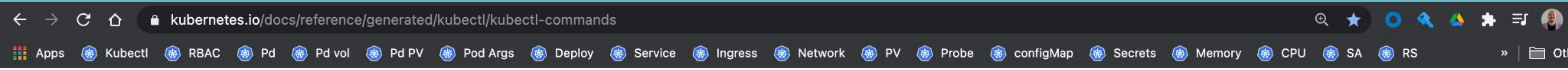
A CLOUD GURU

PLURALSIGHT

$ killer_ Kubernetes Exam Simulator

CloudAcademy

Study Tips - Use bookmarks!



In the exam you are allowed to use bookmarks for the documentation so as you study and learn add bookmarks to the page in kubernetes docs.

When you do your practice tests and practical work, refer to the bookmarks and not your notes, this will help you be fast for exam day.

Study Tips - Use bookmarks!

Study Tips - kubectl Reference Docs

Study Tips - VIM

If you don’t know VIM editor learn it. You can get away with just knowing nano, by setting KUBE_EDITOR=”nano” but when you need to edit deployments etc it will default to VIM so you don’t want to risk being stuck. If you don't know VIM you are not ready for you exam.

VIM Cheat Sheet: vim.rtorr.com

Study Tips - VIM

There are two modes, for writing/deleting text press **i**, then press **esc** for edit mode

To exit without saving press **esc** and type **:q!**
To exit with saving press **esc** and type **:wq!**

To delete a line type **dd**
To delete more than one line type **5dd** (substitute 5 for the number you want to delete)

To Goto line 5 type **5gg** (substitute 5 for the number you want to goto)
To Jump to the end of the line type **$**

To search for a word eg image press **esc** then type **/image** and press **enter**

To show line numbers press **esc** and then type **:set number**

Study Tips - Basic linux commands

If you are a developer doing this course and you are not used to linux, it is worth learning basic linux commands.

Command	Description	Example
cd	Used to go into or out of a folder	**cd myfolder** (**cd ..** to go back)
ls	List files/folders in a directory	**ls** (**ls myfolder** to list contents of myfolder)
grep	Searches for text	**cat 1.yaml | grep image**
cat	Read a file on the console	**cat deployment.yaml**
cp	Copy a file	**cp 1.yaml 1a.yaml**
mv	Move a file	**mv 1.yaml backup/1.yaml**
mkdir	Make a directory	**mkdir test**
rm	Removes a file or folder	**rm 1.yaml** (**rm -R myfolder** for directory)
vi	Go into vi editor	**vi deployment.yaml**

Practical Examples

Practical Examples

- 13% – Core Concepts
- 18% – Configuration
- 10% – Multi-Container Pods
- 18% – Observability
- 20% – Pod Design
- 13% – Services & Networking
- 8% – State Persistence

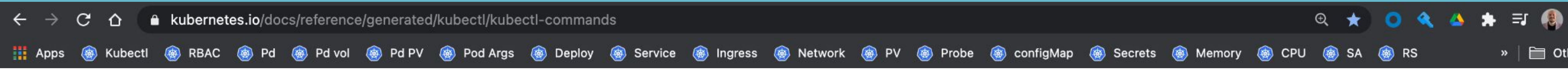
Practical Examples - General

Command	Description
kubectl get pods --show-labels	Show labels for pods
Kubectl get pods -l app=postgres	List pods that have the label app=postgres
kubectl get pods -A	Show pods in all namespaces
kubectl get all -n dev	Show all (most) things in the namespace dev
kubectl api-resources	Get names/short names for resources
kubectl run -h	Get help for kubectl
alias k=kubectl	Sets k as the alias for kubectl

Practical Examples - Core Concepts

```
kubectl run -h
kubectl run nginx --image nginx --dry-run=client -o yaml > 1.yaml


kubectl run hazelcast --image=hazelcast/hazelcast --port=5701

kubectl run hazelcast --image=hazelcast/hazelcast --env="app=hazelcast,env=prod"

kubectl run hazelcast --image=hazelcast/hazelcast --labels="app=hazelcast,env=prod"

kubectl run hazelcast --image=hazelcast/hazelcast --command ‘sleep 3600’

kubectl run hazelcast --image=hazelcast/hazelcast --requests=’cpu=100m,memory=256Mi’
kubectl run hazelcast --image=hazelcast/hazelcast --limits=’cpu=200m,memory=512Mi’
```

Practical Examples - Configuration



# Config Maps (cm for short)

kubectl create configmap myconfig --from-literal=username=brad

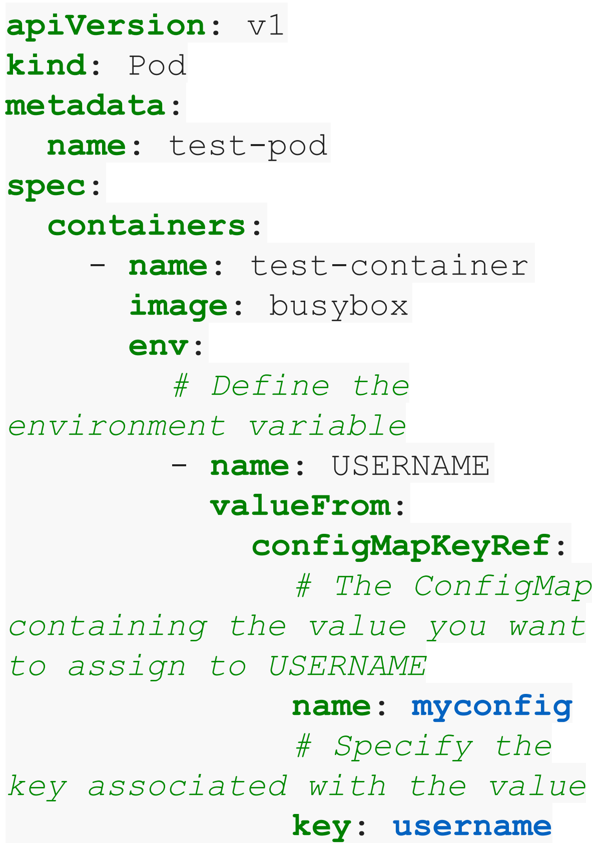
kubectl get configmap myconfig -o yaml

kubectl run test-pod --image busybox --dry-run=client -o yaml > 1.yaml

kubectl create -f 1.yaml

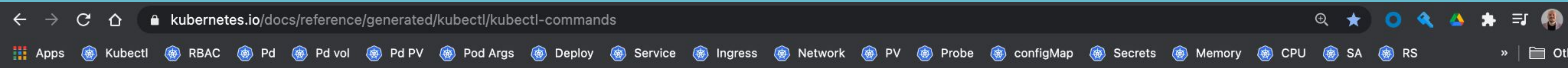
https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#-em-configmap-em-

https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
    - name: test-container
      image: busybox
      env:
        # Define the
environment variable
          - name: USERNAME
            valueFrom:
              configMapKeyRef:
                # The ConfigMap
containing the value you want
to assign to USERNAME
                name: myconfig
                # Specify the
key associated with the value
                key: username
```

Practical Examples - Configuration



kubernetes.io/docs/reference/generated/kubectl/kubectl-commands

# Secrets

kubectl create secret generic mysecret --from-literal=password=123

kubectl get secret mysecret -o yaml

kubectl run test-pod --image busybox --dry-run=client -o yaml > 1.yaml

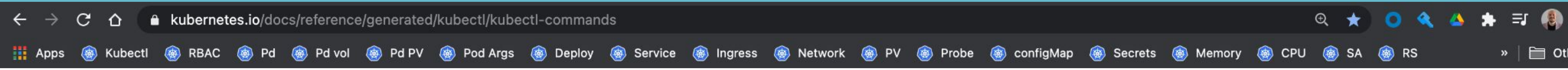
kubectl create -f 1.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
    - name: test-container
      image: busybox
      env:
        # Define the environment variable
        - name: PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: password
```

https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#-em-secret-generic-em-

https://kubernetes.io/docs/concepts/configuration/secret/

Practical Examples - Configuration



kubernetes.io/docs/reference/generated/kubectl/kubectl-commands

```
# Service account

kubectl create serviceaccount my-service-account

kubectl get serviceaccount my-service-account

kubectl run test-pod --image busybox --serviceaccount='my-service-account'
```

https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#-em-serviceaccount-em-

https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/

Practical Examples - Observability

# Get logs from pod
kubectl logs my-pod

# Get logs from pods with the name label myLabel
kubectl logs -l name=myLabel

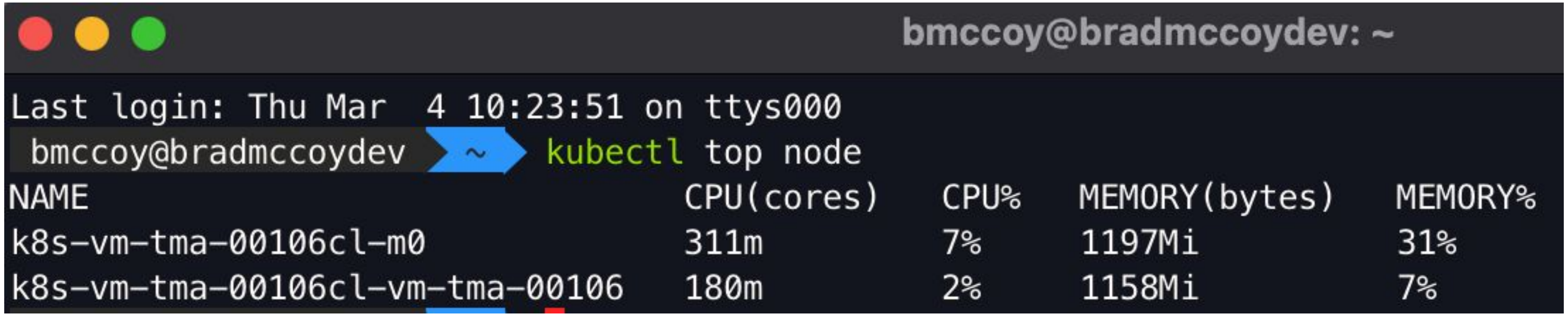
# Output logs to console
kubectl exec my-pod -c my-container -- cat /var/log.txt (multi container)

# Go into the pod to look for the logs
kubectl -n your-namespace exec -it pod/your-pod -- /bin/sh

# Kubectl cheatsheet
https://kubernetes.io/docs/reference/kubectl/cheatsheet

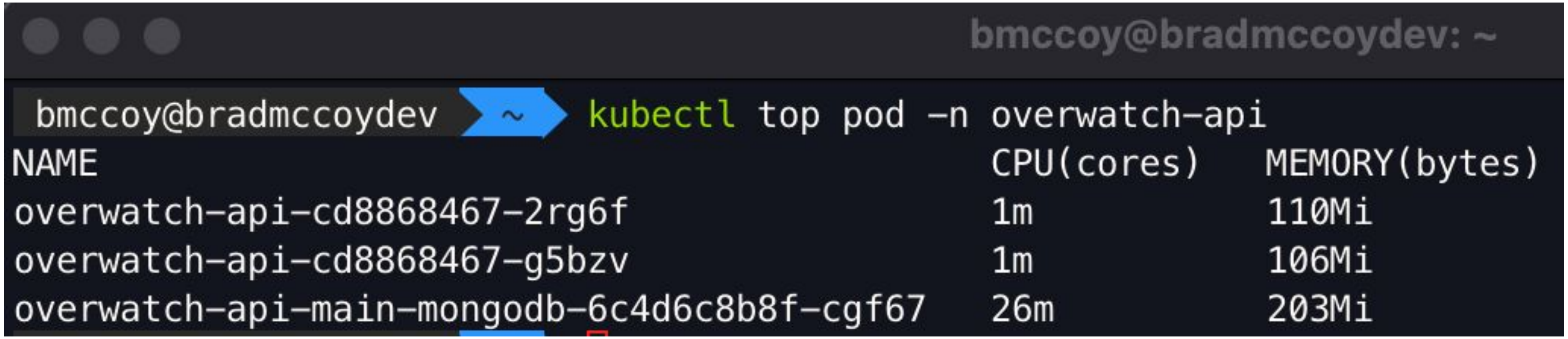
Practical Examples - Observability

kubectl top node



```
Last login: Thu Mar  4 10:23:51 on ttys000
 bmccoy@bradmccoydev  ~  kubectl top node
NAME                              CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
k8s-vm-tma-00106cl-m0             311m         7%     1197Mi          31%
k8s-vm-tma-00106cl-vm-tma-00106   180m         2%     1158Mi          7%
```

kubectl top pod



```
 bmccoy@bradmccoydev  ~  kubectl top pod -n overwatch-api
NAME                                       CPU(cores)   MEMORY(bytes)
overwatch-api-cd8868467-2rg6f              1m           110Mi
overwatch-api-cd8868467-g5bzv              1m           106Mi
overwatch-api-main-mongodb-6c4d6c8b8f-cgf67   26m       203Mi
```

Practical Examples - Pod Design

```
kubectl create deployment my-dep --image=nginx --replicas=3 --port=5701
```

Practical Examples - Pod Design

```
# Rolling updates, and rollbacks

kubectl set image deployment/nginx-deployment nginx=nginx:1.16.1 --record
kubectl rollout status deployment/nginx-deployment
kubectl rollout undo deployment/nginx-deployment

# Jobs
kubectl create job my-job --image=busybox -- date
https://kubernetes.io/docs/concepts/workloads/controllers/job

# Cronjobs
https://kubernetes.io/docs/concepts/workloads/controllers/cron-jobs
```

Practical Examples - Services and Networking

# Services

kubectl create service clusterip my-cs --tcp=5678:8080

kubectl create service nodeport my-ns --tcp=5678:8080

Kubectl expose deployment nginx --port=80 --target-port=8000 (--type=ClusterIP,NodePort)

# Network policies

https://kubernetes.io/docs/concepts/services-networking/network-policies/

Practical Examples - State Persistence

# Host path example

https://kubernetes.io/docs/concepts/storage/volumes

If you can’t find something in the documentation remember you can use the kubectl explain command

kubectl explain pv --recursive

kubectl explain pod --recursive

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: test-pd
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /test-pd
      name: test-volume
  volumes:
  - name: test-volume
    hostPath:
      # directory location on host
      path: /data
      # this field is optional
      type: Directory
```

Practical Examples - State Persistence

We will do a separate talk on this as there is alot to cover

https://kubernetes.io/docs/tasks/configure-pod-container/configure-volume-storage

https://kubernetes.io/docs/tasks/configure-pod-container/configure-persistent-volume-storage

Conclusion

Don’t be afraid to fail. I failed my first attempt but it gave me a good opportunity to see where I needed to improve, and passed on my free retake with a high score.

Learn the basics first, you will save time in the long run, learn basic linux, learn VIM, learn kubectl well.

If you are having trouble getting a test environment to practice with maybe your cpu, ram on your computer can’t run Kubernetes or can’t afford it, reach out to me and I will see what I can do.

Have fun studying, and remember there is a wonderful community out there to help you!

Important Links -

https://github.com/bradmccoydev/cncf-community-manly

Links will be added to: https://github.com/bradmccoydev/cncf-community-manly

https://kubernetes.io/docs/contribute/participate

https://www.cncf.io/certification/ckad

https://kubernetes.io/docs/home

https://itnext.io/kubernetes-journey-cka-ckad-exam-tips-ff73e4672833

https://events.linuxfoundation.org/kubecon-cloudnativecon-europe/register

slack.k8s.io

slack.cncf.io

vim.rtorr.com

CLOUD NATIVE
COMPUTING
FOUNDATION

Thank You

Join us: https://community.cncf.io/manly

CC-BY License