< Question 12 >

Task weight: 6%

Use context: `kubectl config use-context k8s-c1-H`

Use *Namespace* `project-tiger` for the following. Create a *Deployment* named `deploy-important` with label `id=very-important` (the `Pods` should also have this label) and 3 replicas. It should contain two containers, the first named container1 with image `nginx:1.17.6-alpine` and the second one named container2 with image `kubernetes/pause`.

There should be only ever **one** *Pod* of that *Deployment* running on **one** worker node. We have two worker nodes: cluster1-worker1 and cluster1-worker2. Because the *Deployment* has three replicas the result should be that on both nodes **one** *Pod* is running. The third *Pod* won't be scheduled, unless a new worker node will be added.

In a way we kind of simulate the behaviour of a *DaemonSet* here, but using a *Deployment* and a fixed number of replicas.

41 minutes

< Question 13 >

Task weight: 4%

Use context: `kubectl config use-context k8s-c1-H`

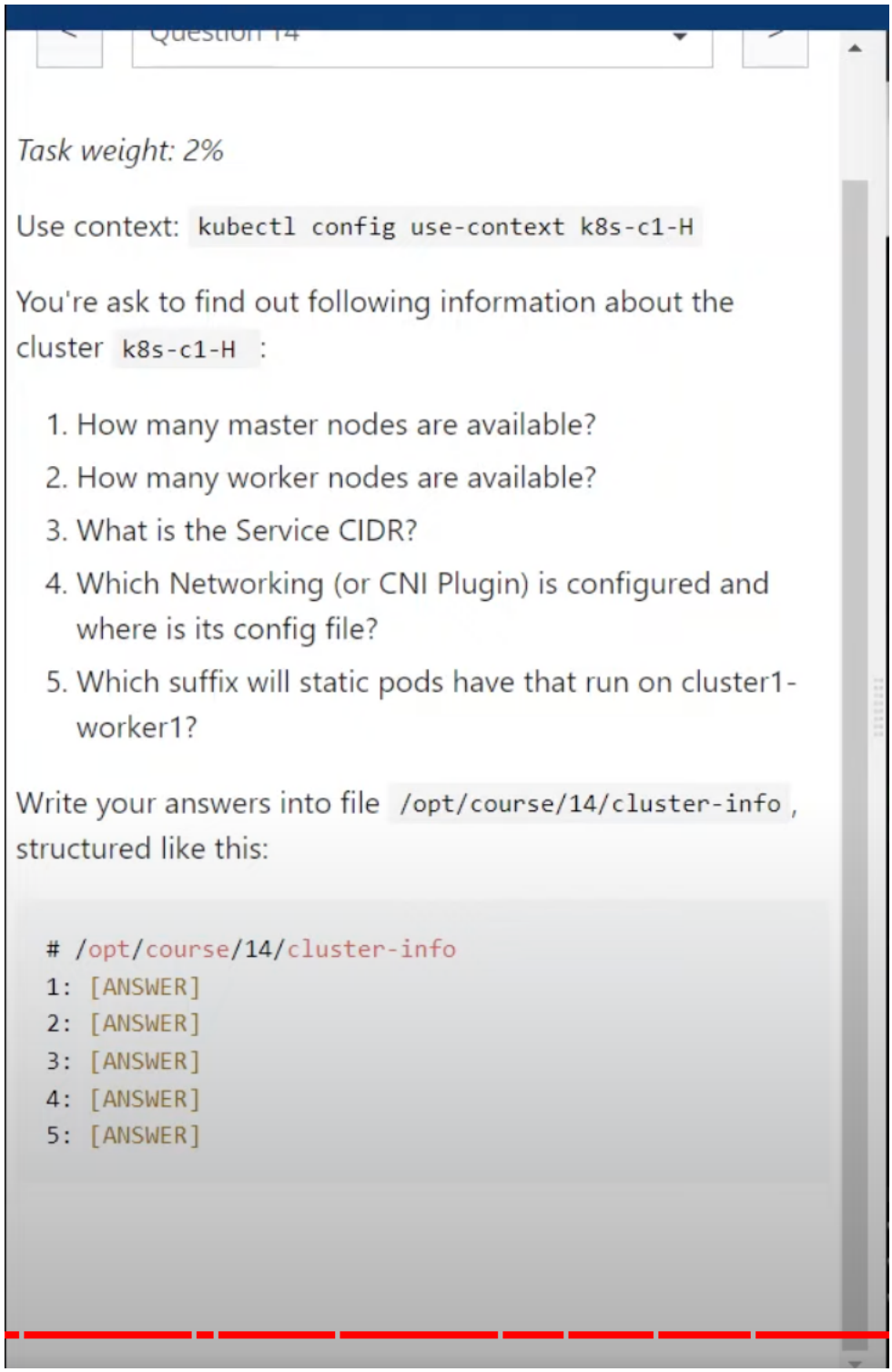
Create a *Pod* named `multi-container-playground` in *Namespace* `default` with three containers, named `c1`, `c2` and `c3`. There should be a volume attached to that *Pod* and mounted into every container, but the volume shouldn't be persisted or shared with other *Pods*.

Container `c1` should be of image `nginx:1.17.6-alpine` and have the name of the node where its *Pod* is running available as environment variable MY_NODE_NAME.

Container `c2` should be of image `busybox:1.31.1` and write the output of the `date` command every second in the shared volume into file `date.log`. You can use `while true; do date >> /your/vol/path/date.log; sleep 1; done` for this.

Container `c3` should be of image `busybox:1.31.1` and constantly send the content of file `date.log` from the shared volume to stdout. You can use `tail -f /your/vol/path/date.log` for this.

Check the logs of container `c3` to confirm correct setup.


Question 14
Task weight: 2%
Use context: kubectl config use-context k8s-c1-H
You're ask to find out following information about the cluster k8s-c1-H :
1. How many master nodes are available?
2. How many worker nodes are available?
3. What is the Service CIDR?
4. Which Networking (or CNI Plugin) is configured and where is its config file?
5. Which suffix will static pods have that run on cluster1-worker1?
Write your answers into file /opt/course/14/cluster-info, structured like this:
/opt/course/14/cluster-info
1: [ANSWER]
2: [ANSWER]
3: [ANSWER]
4: [ANSWER]
5: [ANSWER]

18 minutes

< Question 15 >

Task weight: 3%

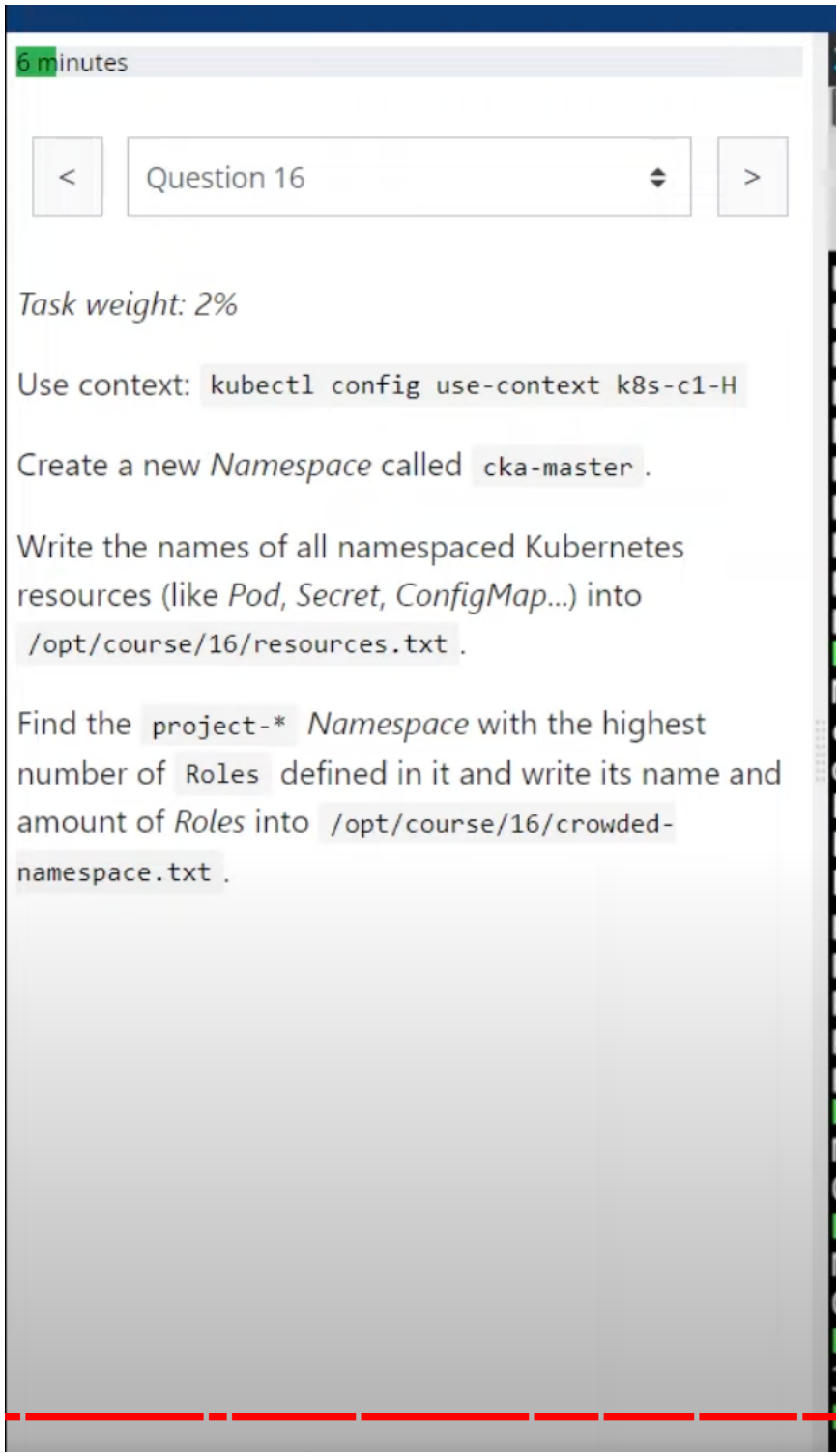
Use context: `kubectl config use-context k8s-c2-AC`

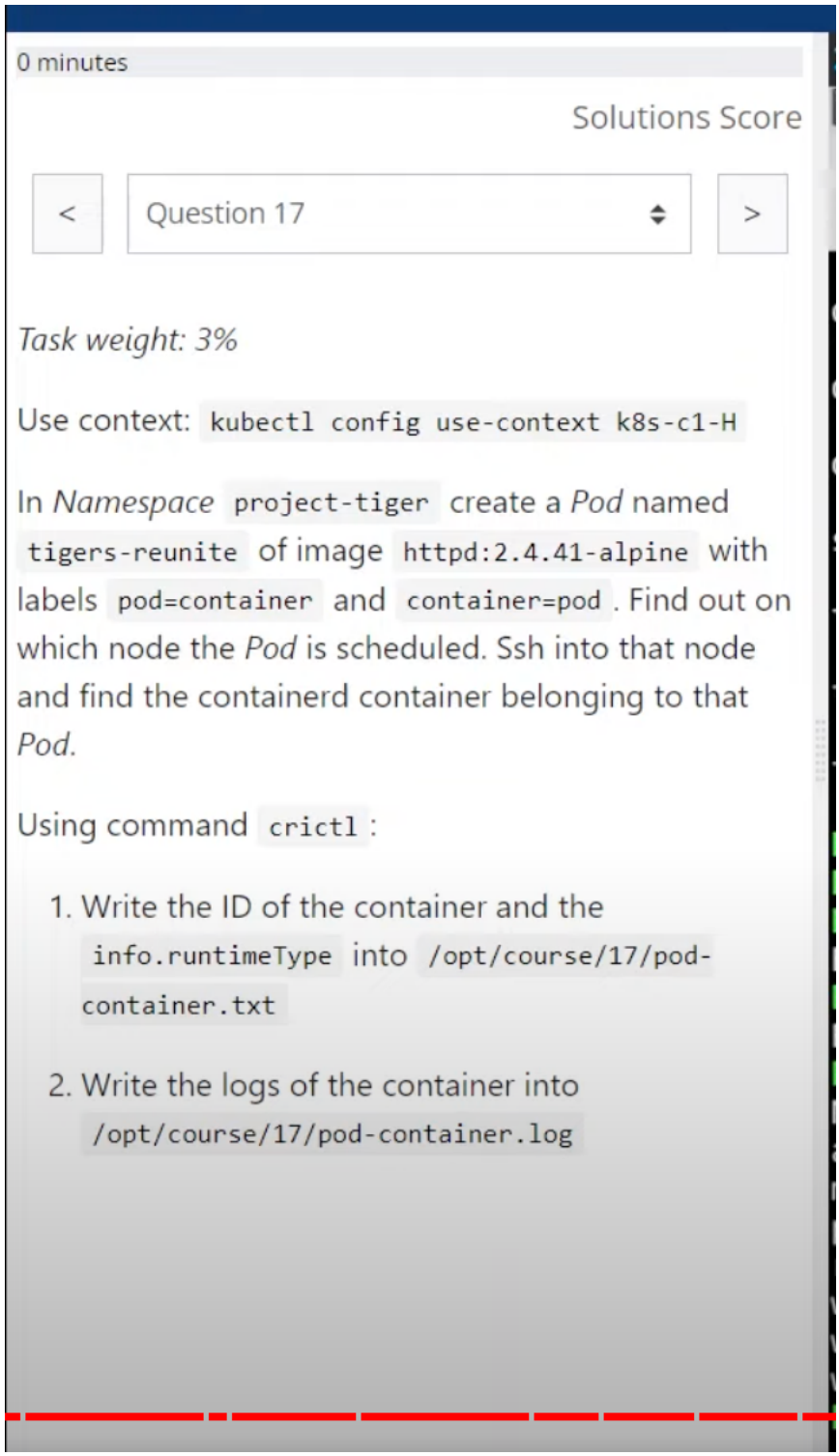
Write a command into `/opt/course/15/cluster_events.sh` which shows the latest events in the whole cluster, ordered by time. Use `kubectl` for it.

Now kill the kube-proxy *Pod* running on node cluster2-worker1 and write the events this caused into `/opt/course/15/pod_kill.log`.

Finally kill the containerd container of the kube-proxy *Pod* on node cluster2-worker1 and write the events into `/opt/course/15/container_kill.log`.

Do you notice differences in the events both actions caused?


6 minutes
<
Question 16
>
Task weight: 2%
Use context: kubectl config use-context k8s-c1-H
Create a new Namespace called cka-master.
Write the names of all namespaced Kubernetes resources (like Pod, Secret, ConfigMap...) into /opt/course/16/resources.txt.
Find the project-* Namespace with the highest number of Roles defined in it and write its name and amount of Roles into /opt/course/16/crowded-namespace.txt.


0 minutes
Solutions Score
<
Question 17
>
Task weight: 3%
Use context: kubectl config use-context k8s-c1-H
In Namespace project-tiger create a Pod named tigers-reunite of image httpd:2.4.41-alpine with labels pod=container and container=pod . Find out on which node the Pod is scheduled. Ssh into that node and find the containerd container belonging to that Pod.
Using command crictl :
1. Write the ID of the container and the info.runtimeType into /opt/course/17/pod-container.txt
2. Write the logs of the container into /opt/course/17/pod-container.log

0 minutes

Solutions Score

< Question 18 >

Task weight: 8%

Use context: `kubectl config use-context k8s-c3-CCC`

There seems to be an issue with the kubelet not running on `cluster3-worker1`. Fix it and confirm that cluster has node `cluster3-worker1` available in Ready state afterwards. You should be able to schedule a *Pod* on `cluster3-worker1` afterwards.

Write the reason of the issue into `/opt/course/18/reason.txt`.

0 minutes

Solutions Score

< Question 19 >

Task weight: 3%

> **NOTE:** This task can only be solved if questions 18 or 20 have been successfully implemented and the k8s-c3-CCC cluster has a functioning worker node

Use context: `kubectl config use-context k8s-c3-CCC`

Do the following in a new *Namespace* `secret`. Create a *Pod* named `secret-pod` of image `busybox:1.31.1` which should keep running for some time.

There is an existing *Secret* located at `/opt/course/19/secret1.yaml`, create it in the *Namespace* `secret` and mount it readonly

0 minutes

Solutions Score

< Question 21 >

Task weight: 2%

Use context: `kubectl config use-context k8s-c3-CCC`

Create a `Static Pod` named `my-static-pod` in *Namespace* `default` on cluster3-master1. It should be of image `nginx:1.16-alpine` and have resource requests for `10m` CPU and `20Mi` memory.

Then create a NodePort *Service* named `static-pod-service` which exposes that static *Pod* on port 80 and check if it has *Endpoints* and if its reachable through the `cluster3-master1` internal IP address. You can connect to the internal node IPs from your main terminal.

0 minutes

Solutions Score

< Question 22 >

Task weight: 2%

Use context: `kubectl config use-context k8s-c2-AC`

Check how long the kube-apiserver server certificate is valid on `cluster2-master1`. Do this with openssl or cfssl. Write the exipiration date into `/opt/course/22/expiration`.

Also run the correct `kubeadm` command to list the expiration dates and confirm both methods show the same date.

Write the correct `kubeadm` command that would renew the apiserver server certificate into `/opt/course/22/kubeadm-renew-certs.sh`.

0 minutes

Solutions Score

< Question 23 >

Task weight: 2%

Use context: `kubectl config use-context k8s-c2-AC`

Node cluster2-worker1 has been added to the cluster using `kubeadm` and TLS bootstrapping.

Find the "Issuer" and "Extended Key Usage" values of the cluster2-worker1:

1. kubelet **client** certificate, the one used for outgoing connections to the kube-apiserver.
2. kubelet **server** certificate, the one used for incoming connections from the kube-apiserver.

Write the information into file `/opt/course/23/certificate-info.txt`.

Compare the "Issuer" and "Extended Key Usage" fields of both certificates and make sense of these.

0 minutes

Solutions Score

< Question 24 >

Task weight: 9%

Use context: `kubectl config use-context k8s-c1-H`

There was a security incident where an intruder was able to access the whole cluster from a single hacked backend *Pod*.

To prevent this create a *NetworkPolicy* called `np-backend` in *Namespace* `project-snake`. It should allow the `backend-*` *Pods* only to:

- connect to `db1-*` *Pods* on port 1111
- connect to `db2-*` *Pods* on port 2222

Use the `app` label of *Pods* in your policy.

After implementation, connections from `backend-*` *Pods* to `vault-*` *Pods* on port 3333 should for example no longer work.

0 minutes

Solutions Score

< Question 25 >

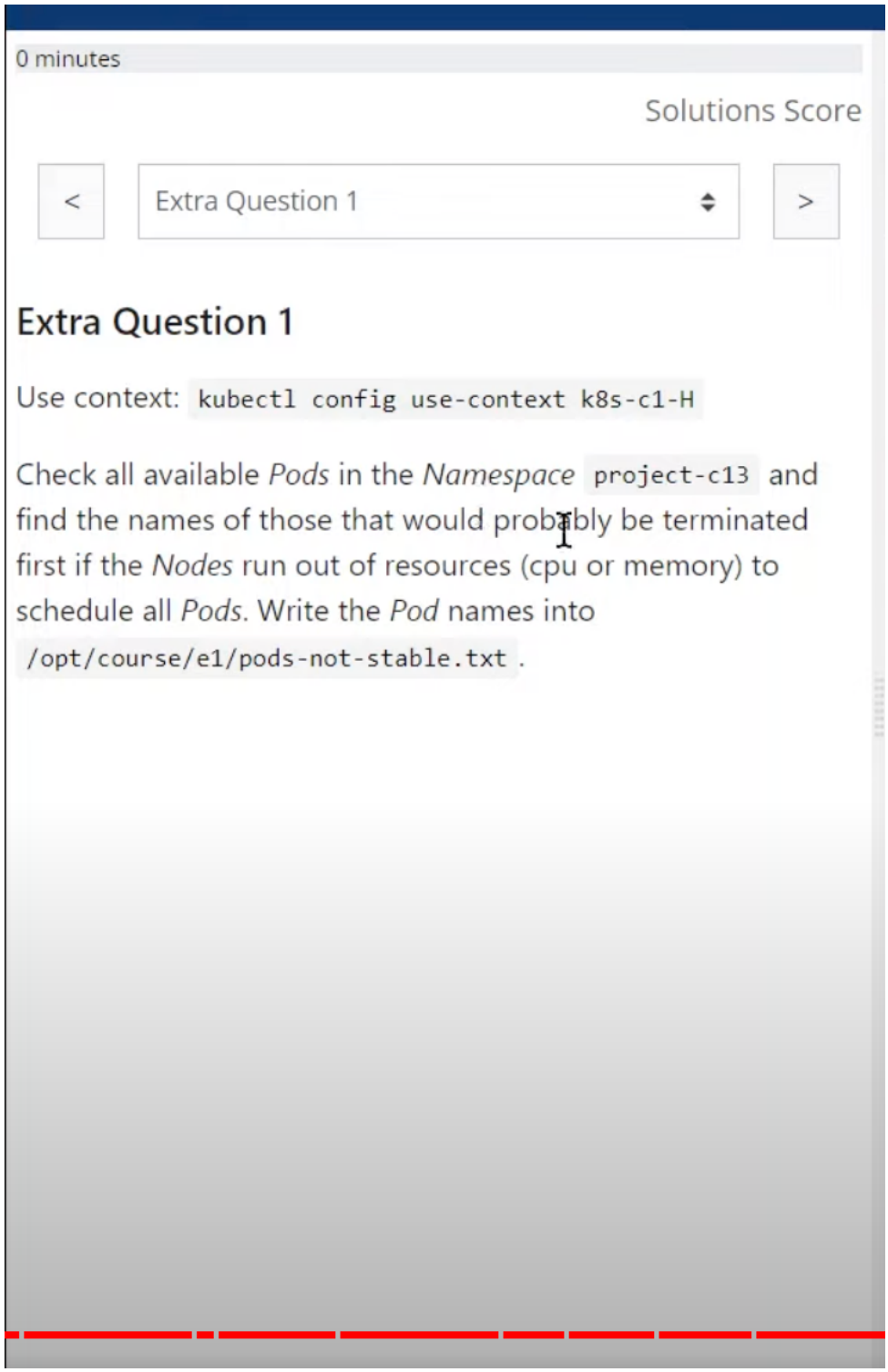
Task weight: 8%

Use context: `kubectl config use-context k8s-c3-CCC`

Make a backup of etcd running on cluster3-master1 and save it on the master node at `/tmp/etcd-backup.db`.

Then create a *Pod* of your kind in the cluster.

Finally restore the backup, confirm the cluster is still working and that the created *Pod* is no longer with us.


0 minutes
Solutions Score
<
Extra Question 1
>
Extra Question 1
Use context: kubectl config use-context k8s-c1-H
Check all available Pods in the Namespace project-c13 and find the names of those that would probably be terminated first if the Nodes run out of resources (cpu or memory) to schedule all Pods. Write the Pod names into /opt/course/e1/pods-not-stable.txt .