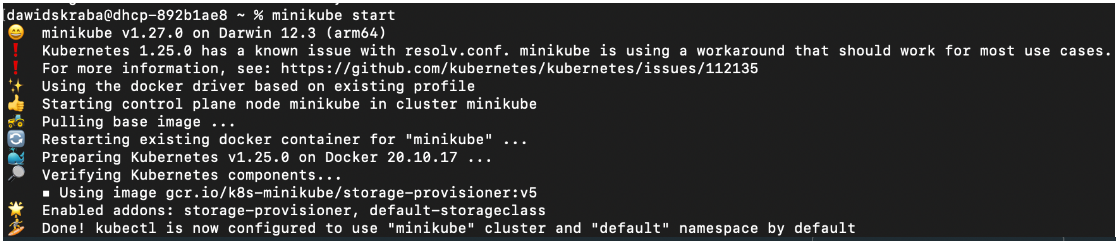
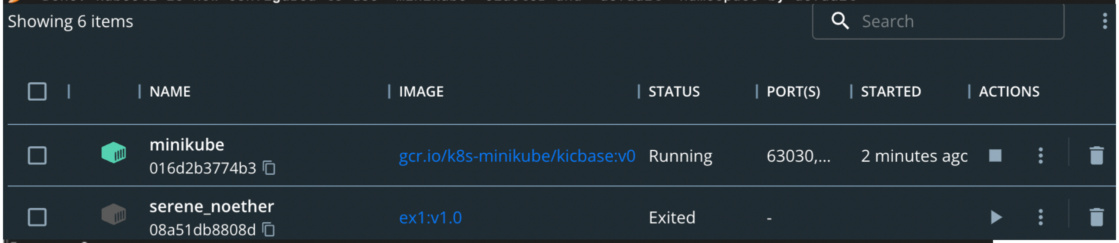
Practical 3

Dawid Skraba 19433692

1. I installed minikube using the docker version. This creates a container which is effectively our node where we will place pods on.

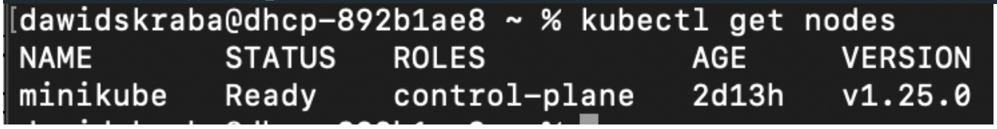


```
[dawidskraba@dhcp-892b1ae8 ~ % minikube start
😄  minikube v1.27.0 on Darwin 12.3 (arm64)
❗  Kubernetes 1.25.0 has a known issue with resolv.conf. minikube is using a workaround that should work for most use cases.
❗  For more information, see: https://github.com/kubernetes/kubernetes/issues/112135
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.25.0 on Docker 20.10.17 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```



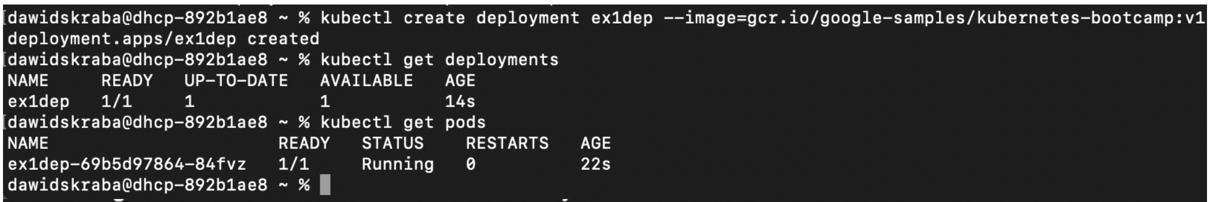
Showing 6 items

□	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
□	minikube 016d2b3774b3	gcr.io/k8s-minikube/kicbase:v0	Running	63030,...	2 minutes agc	■ ⋮ 🗑
□	serene_noether 08a51db8808d	ex1:v1.0	Exited	-		▶ ⋮ 🗑

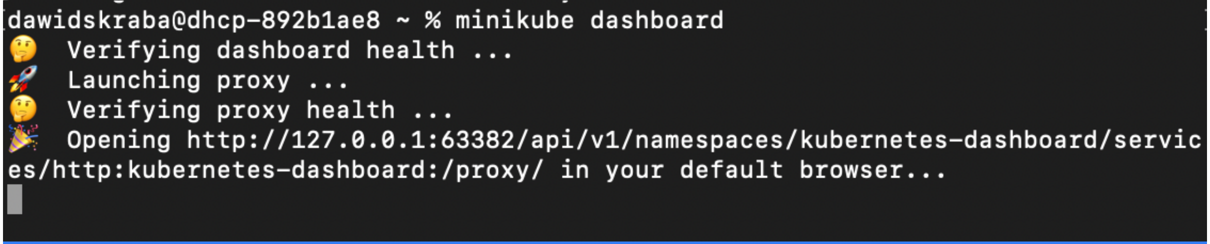


```
[dawidskraba@dhcp-892b1ae8 ~ % kubectl get nodes
 NAME       STATUS   ROLES           AGE     VERSION
 minikube   Ready    control-plane   2d13h   v1.25.0
```

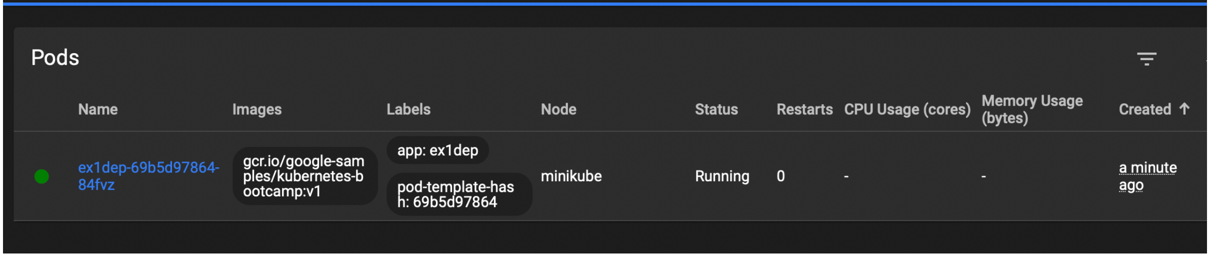
2.



```
[dawidskraba@dhcp-892b1ae8 ~ % kubectl create deployment ex1dep --image=gcr.io/google-samples/kubernetes-bootcamp:v1
deployment.apps/ex1dep created
[dawidskraba@dhcp-892b1ae8 ~ % kubectl get deployments
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
ex1dep   1/1     1            1           14s
[dawidskraba@dhcp-892b1ae8 ~ % kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
ex1dep-69b5d97864-84fvz   1/1     Running   0          22s
dawidskraba@dhcp-892b1ae8 ~ %
```



```
[dawidskraba@dhcp-892b1ae8 ~ % minikube dashboard
🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
🎉  Opening http://127.0.0.1:63382/api/v1/namespaces/kubernetes-dashboard/servic
es/http:kubernetes-dashboard:/proxy/ in your default browser...
```



Pods

	Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
●	ex1dep-69b5d97864-84fvz	gcr.io/google-samples/kubernetes-bootcamp:v1	app: ex1dep pod-template-hash: 69b5d97864	minikube	Running	0	-	-	a minute ago

Using the kubectl create deployment command we create a deployment which we call "ex1dep". Using the –image tag we pull an image from the web. This command will pull this image into a container then onto a pod and finally onto our single node(worker node). To view this I used the command "kubectl get deployments" to view our deployment which is up and running. Then I used the command "kubect get pods" to view our pod we created. To check the clusters runtime we can use the above commands, but we can also use the command "minikube dashboard" to view a dashboard which we can use to monitor our cluster in the browser.