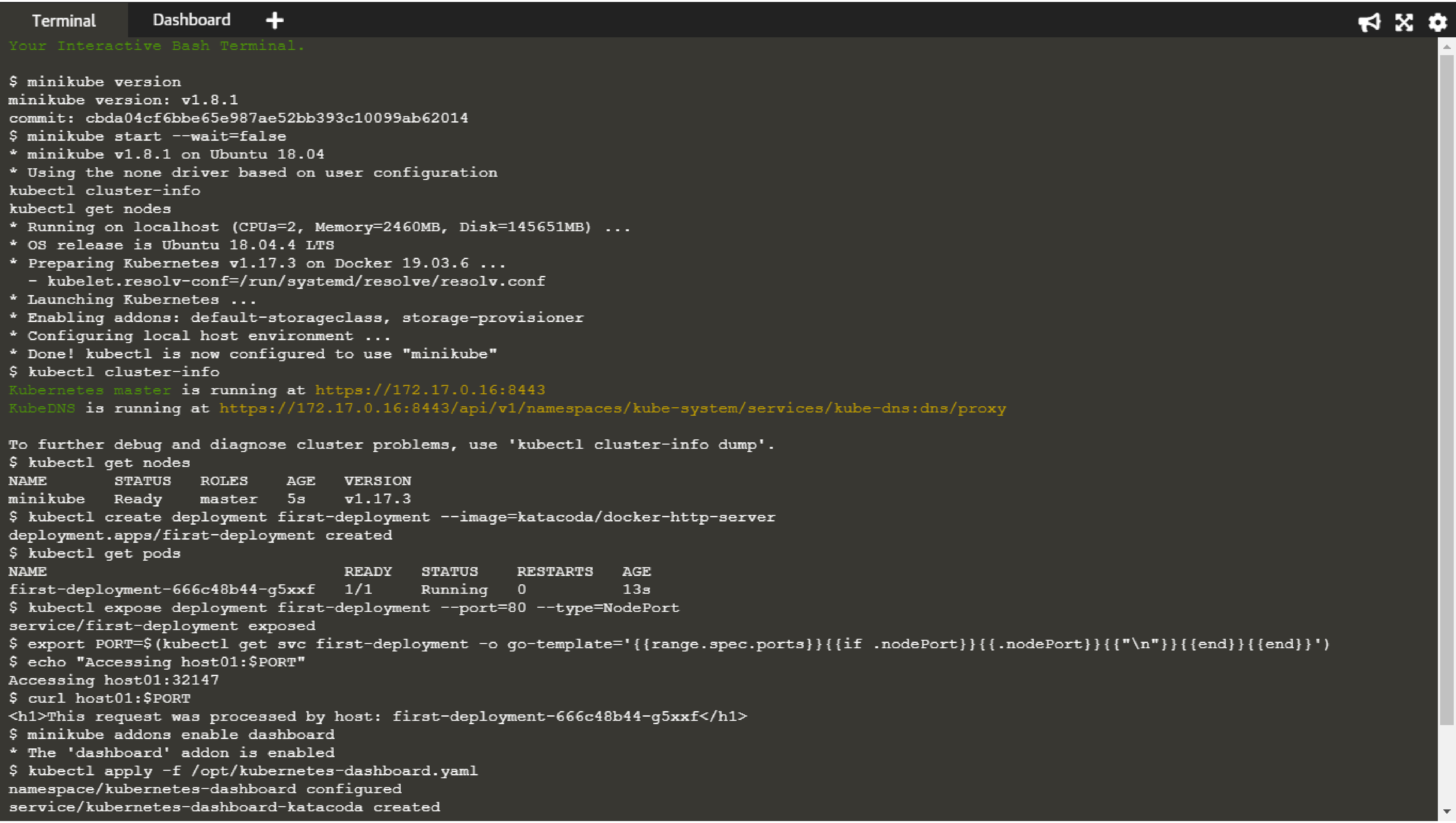
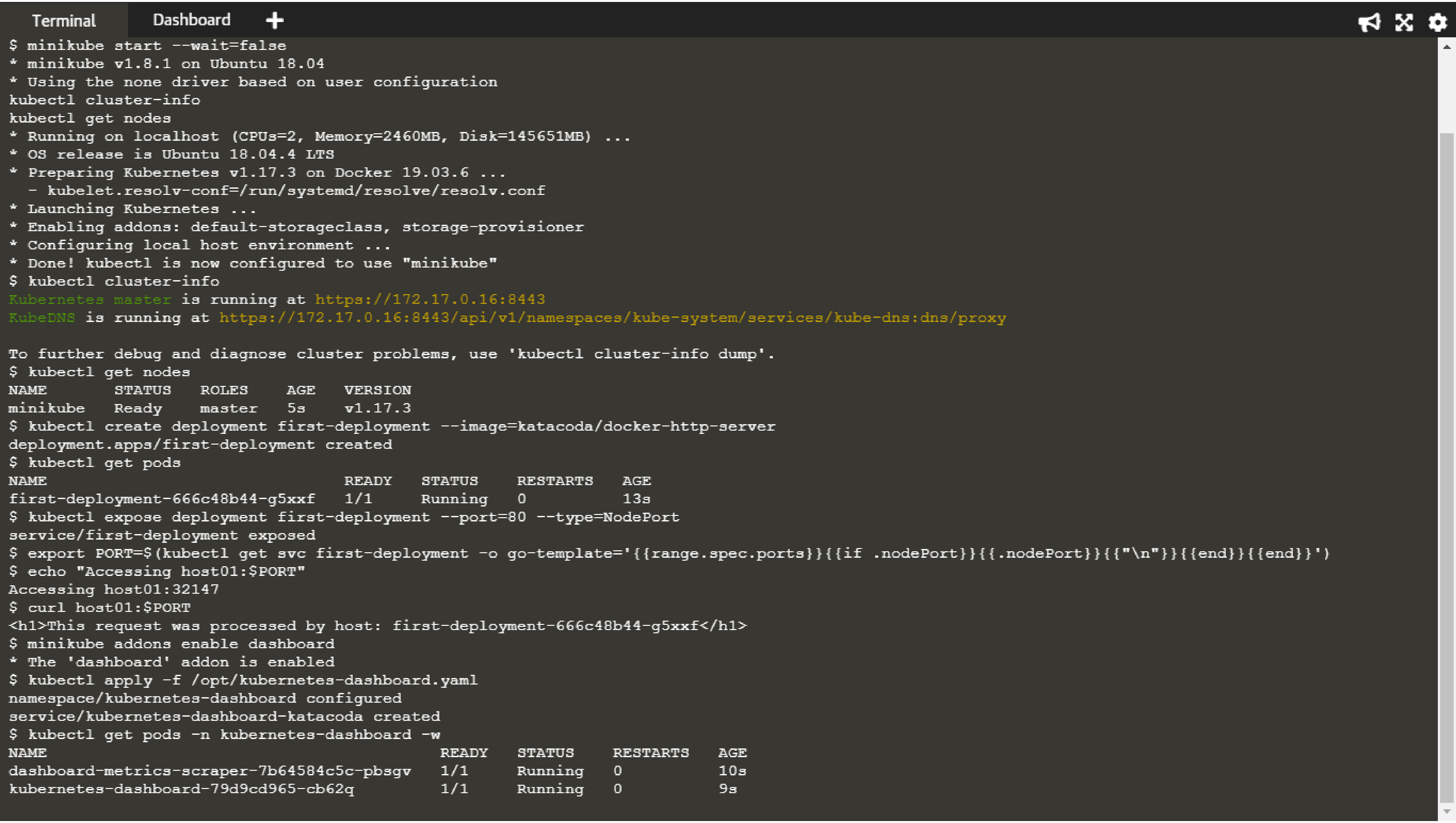
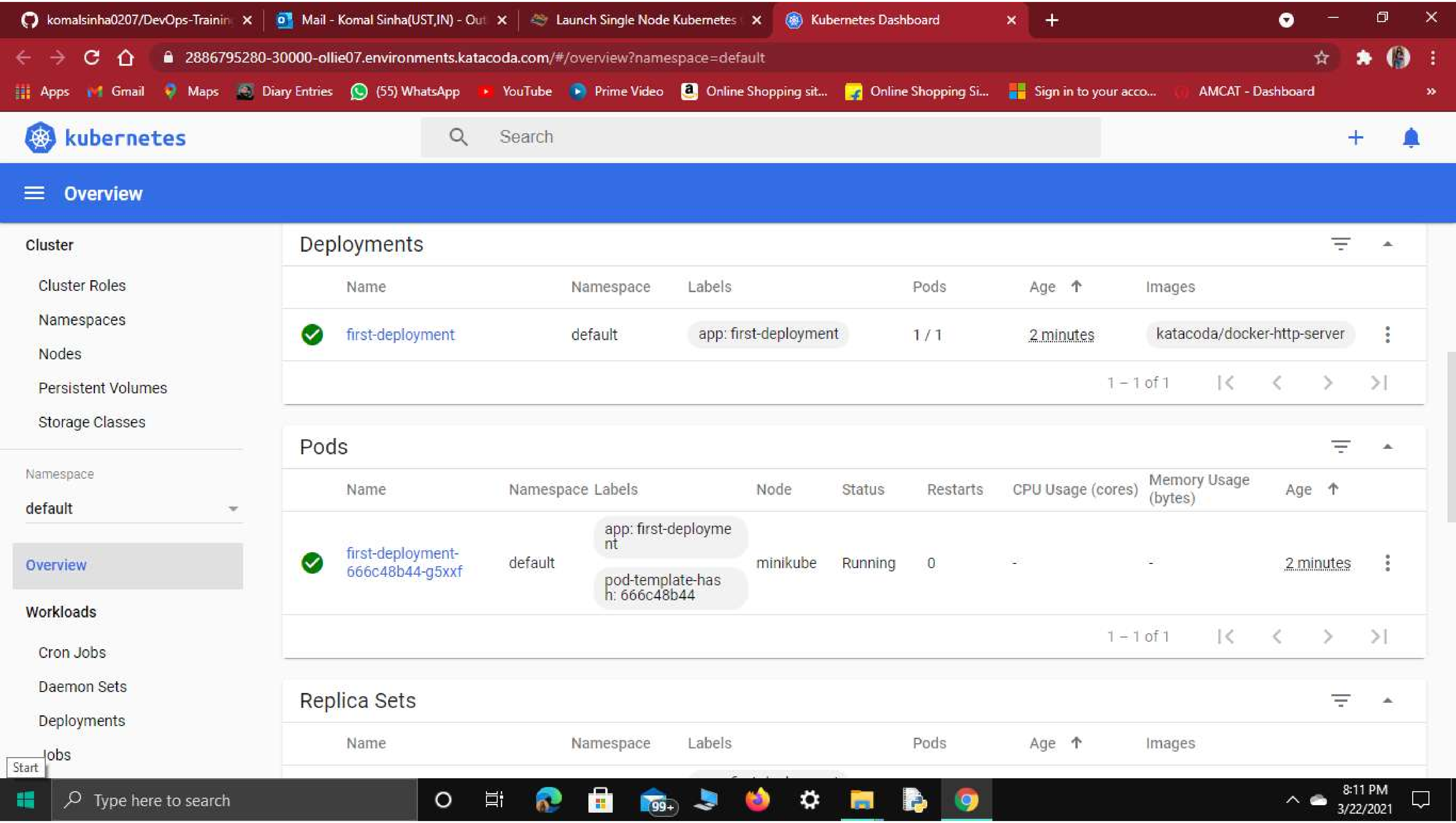


```
Your Interactive Bash Terminal.

$ minikube version
minikube version: v1.8.1
commit: cbda04cf6bbe65e987ae52bb393c10099ab62014
$ minikube start --wait=false
* minikube v1.8.1 on Ubuntu 18.04
* Using the none driver based on user configuration
kubectl cluster-info
kubectl get nodes
* Running on localhost (CPUs=2, Memory=2460MB, Disk=145651MB) ...
* OS release is Ubuntu 18.04.4 LTS
* Preparing Kubernetes v1.17.3 on Docker 19.03.6 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Launching Kubernetes ...
* Enabling addons: default-storageclass, storage-provisioner
* Configuring local host environment ...
* Done! kubectl is now configured to use "minikube"
$ kubectl cluster-info
Kubernetes master is running at https://172.17.0.16:8443
KubeDNS is running at https://172.17.0.16:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
$ kubectl get nodes
NAME       STATUS   ROLES    AGE   VERSION
minikube   Ready    master   5s    v1.17.3
$ kubectl create deployment first-deployment --image=katacoda/docker-http-server
deployment.apps/first-deployment created
$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
first-deployment-666c48b44-g5xxf   1/1     Running   0          13s
$ kubectl expose deployment first-deployment --port=80 --type=NodePort
service/first-deployment exposed
$ export PORT=$(kubectl get svc first-deployment -o go-template='{{range.spec.ports}}{{if .nodePort}}{{.nodePort}}{{"\n"}}{{end}}{{end}}')
$ echo "Accessing host01:$PORT"
Accessing host01:32147
$ curl host01:$PORT
<h1>This request was processed by host: first-deployment-666c48b44-g5xxf</h1>
$ minikube addons enable dashboard
* The 'dashboard' addon is enabled
$ kubectl apply -f /opt/kubernetes-dashboard.yaml
namespace/kubernetes-dashboard configured
service/kubernetes-dashboard-katacoda created
```



```
$ minikube start --wait=false
* minikube v1.8.1 on Ubuntu 18.04
* Using the none driver based on user configuration
kubectl cluster-info
kubectl get nodes
* Running on localhost (CPUs=2, Memory=2460MB, Disk=145651MB) ...
* OS release is Ubuntu 18.04.4 LTS
* Preparing Kubernetes v1.17.3 on Docker 19.03.6 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Launching Kubernetes ...
* Enabling addons: default-storageclass, storage-provisioner
* Configuring local host environment ...
* Done! kubectl is now configured to use "minikube"
$ kubectl cluster-info
Kubernetes master is running at https://172.17.0.16:8443
KubeDNS is running at https://172.17.0.16:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
$ kubectl get nodes
NAME       STATUS   ROLES    AGE   VERSION
minikube   Ready    master   5s    v1.17.3
$ kubectl create deployment first-deployment --image=katacoda/docker-http-server
deployment.apps/first-deployment created
$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
first-deployment-666c48b44-g5xxf   1/1     Running   0          13s
$ kubectl expose deployment first-deployment --port=80 --type=NodePort
service/first-deployment exposed
$ export PORT=$(kubectl get svc first-deployment -o go-template='{{range.spec.ports}}{{if .nodePort}}{{.nodePort}}{{"\n"}}{{end}}{{end}}')
$ echo "Accessing host01:$PORT"
Accessing host01:32147
$ curl host01:$PORT
<h1>This request was processed by host: first-deployment-666c48b44-g5xxf</h1>
$ minikube addons enable dashboard
* The 'dashboard' addon is enabled
$ kubectl apply -f /opt/kubernetes-dashboard.yaml
namespace/kubernetes-dashboard configured
service/kubernetes-dashboard-katacoda created
$ kubectl get pods -n kubernetes-dashboard -w
NAME                                         READY   STATUS    RESTARTS   AGE
dashboard-metrics-scraper-7b64584c5c-pbsgv   1/1     Running   0          10s
kubernetes-dashboard-79d9cd965-cb62q         1/1     Running   0          9s
```



Deployments

	Name	Namespace	Labels	Pods	Age ↑	Images
✓	first-deployment	default	app: first-deployment	1 / 1	2 minutes	katacoda/docker-http-server

1 – 1 of 1

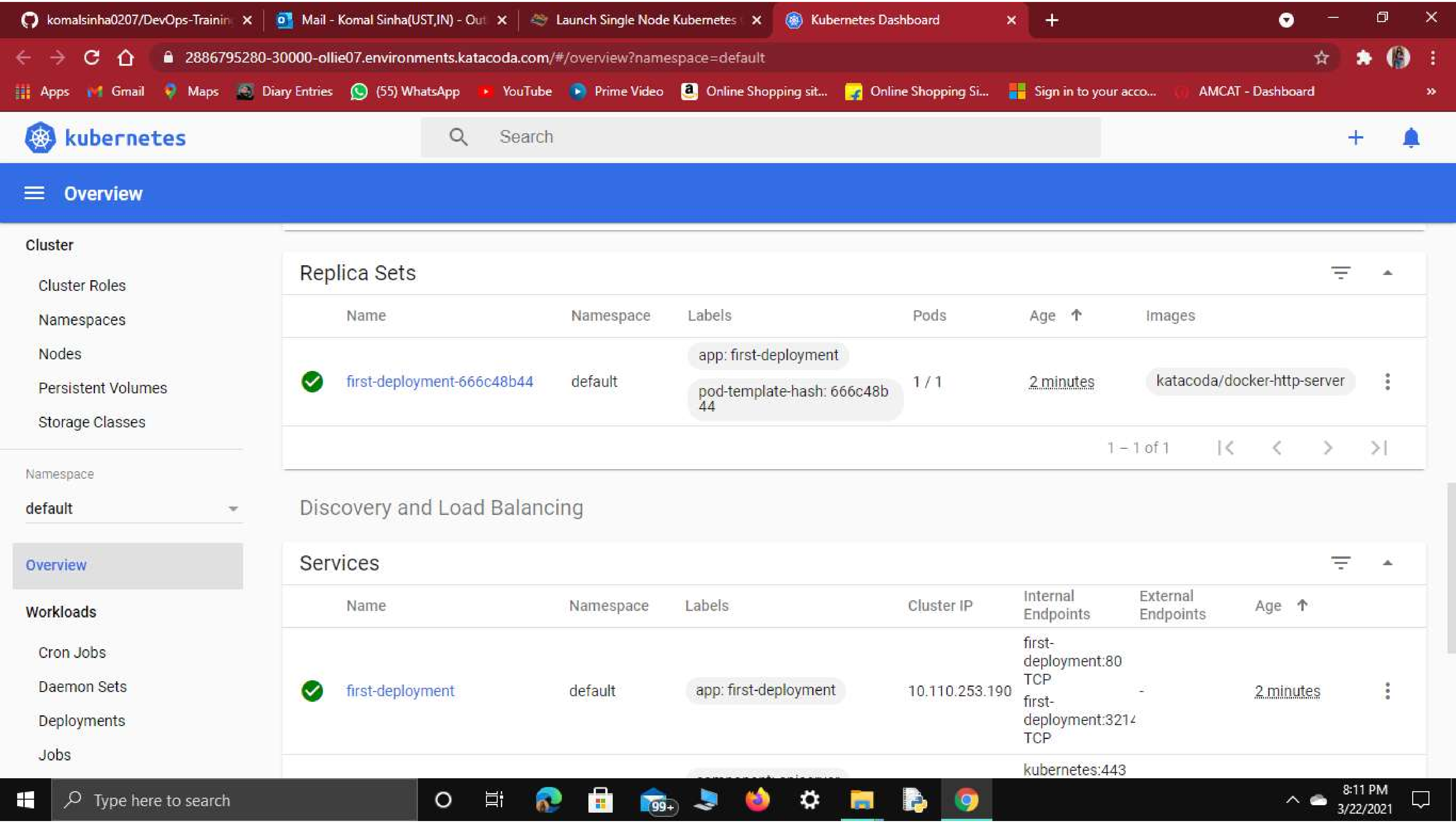
Pods

	Name	Namespace	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Age ↑
✓	first-deployment-666c48b44-g5xxf	default	app: first-deployment pod-template-hash: 666c48b44	minikube	Running	0	-	-	2 minutes

1 – 1 of 1

Replica Sets

Name	Namespace	Labels	Pods	Age ↑	Images



Replica Sets

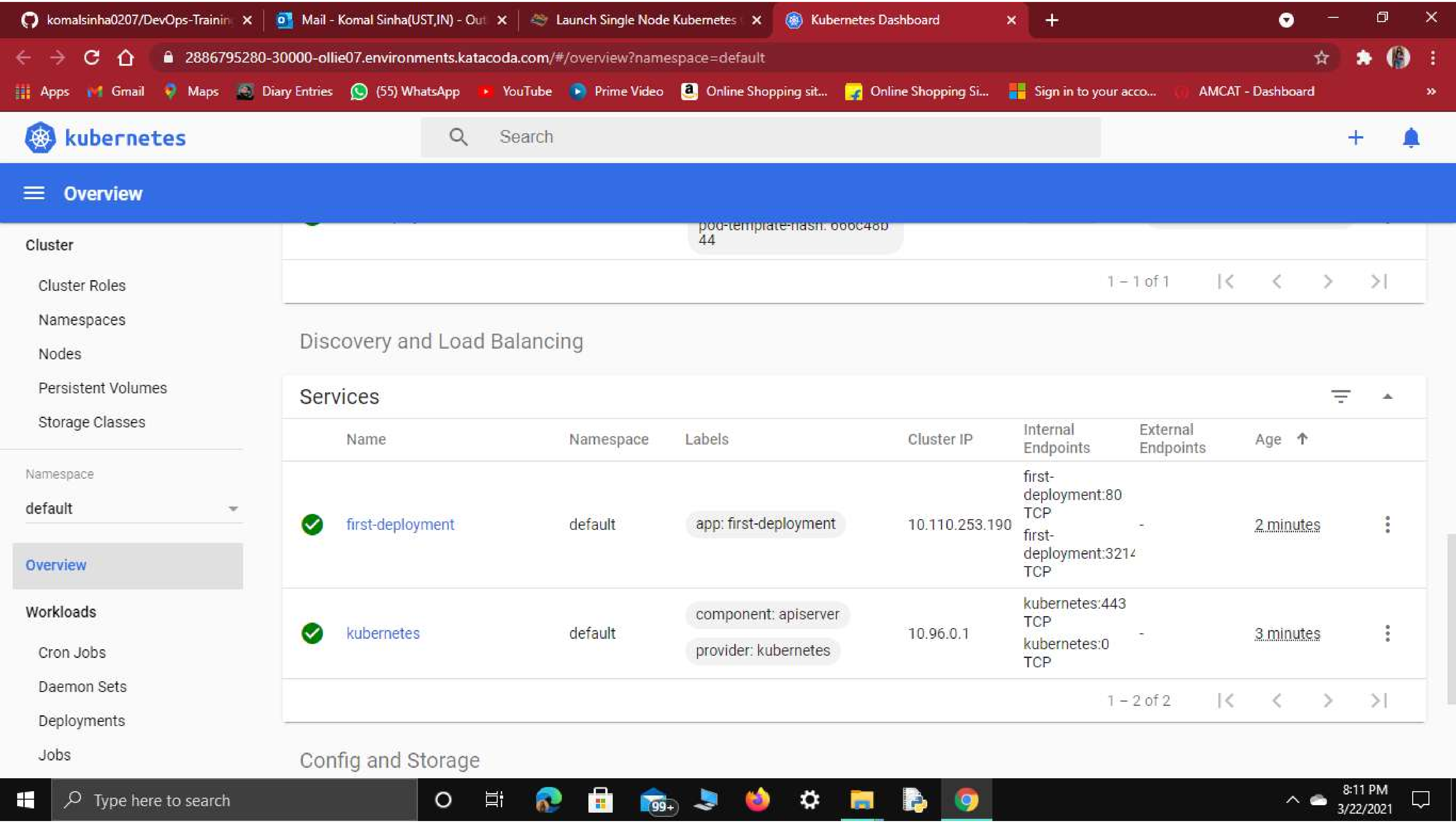
	Name	Namespace	Labels	Pods	Age ↑	Images
✓	first-deployment-666c48b44	default	app: first-deployment pod-template-hash: 666c48b44	1 / 1	2 minutes	katacoda/docker-http-server

1 – 1 of 1

Discovery and Load Balancing

Services

	Name	Namespace	Labels	Cluster IP	Internal Endpoints	External Endpoints	Age ↑
✓	first-deployment	default	app: first-deployment	10.110.253.190	first-deployment:80 TCP first-deployment:3214 TCP	-	2 minutes
			component: apiserver		kubernetes:443		



pod-template-hash: 666c48b
44

1 – 1 of 1

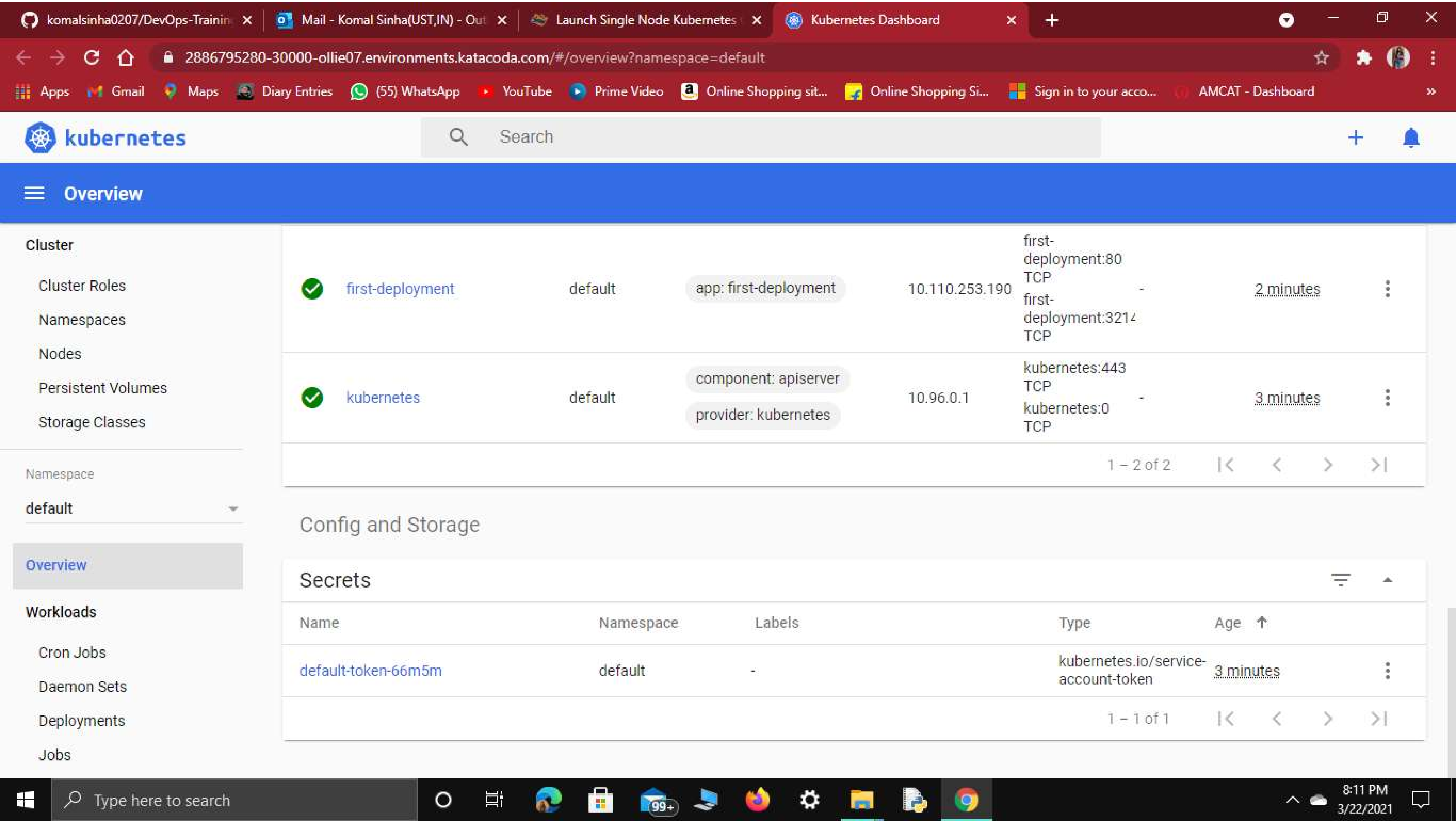
Discovery and Load Balancing

Services

	Name	Namespace	Labels	Cluster IP	Internal Endpoints	External Endpoints	Age ↑
✓	first-deployment	default	app: first-deployment	10.110.253.190	first-deployment:80 TCP first-deployment:3214 TCP	-	2 minutes
✓	kubernetes	default	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	3 minutes

1 – 2 of 2

Config and Storage



✓	first-deployment	default	app: first-deployment	10.110.253.190	first-deployment:80 TCP first-deployment:3214 TCP	-	2 minutes
✓	kubernetes	default	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	3 minutes

1 – 2 of 2

Config and Storage

Secrets

Name	Namespace	Labels	Type	Age ↑
default-token-66m5m	default	-	kubernetes.io/service-account-token	3 minutes

1 – 1 of 1