

- Containers
- Images
- Volumes
- Dev Environments
- Extensions
- Add Extensions

Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another.

Only show running containers

Search

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
	k8s_flask-app_flask-app-5989b67fb7-6m4kd_defa	customer_care_registry1:latest	Running		2 hours ago	
	k8s_flask-app_flask-app-5989b67fb7-jbrfM_kuberr	customer_care_registry1:latest	Running		2 hours ago	
	k8s_kubernetes-dashboard_kubernetes-dashboa	kubernetes/dashboard:v2.6.1	Running		3 hours ago	
	k8s_dashboard-metrics-scraper_dashboard-metr	kubernetes/metrics-scraper:v1.0.8	Running		3 hours ago	
	hardcore_tharp	flink:latest	Exited	5000:5000		
	eager_goodall	gunicorn:latest	Exited (2)			
	zen_robinson	customer_care_registry1:latest	Exited	5000:5000		



RAM 3.35GB CPU 1.69% Connected to Hub

Showing 7 items

4.14.0

```

D:\final_deliver>kubecttl get pods
NAME                                READY   STATUS    RESTARTS   AGE
customer-care-registry-57q7q58877-xz9dx  1/1     Running   0           12m
flask-app-798b5b7dcd-lfcd6             1/1     Running   0           4h59m

D:\final_deliver>kubecttl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE    IP              NODE              NOMINATED NODE   READINESS GATES
customer-care-registry-57q7q58877-xz9dx  1/1     Running   0           12m    172.38.232.287  10.104.182.217   <none>           <none>
flask-app-798b5b7dcd-lfcd6             1/1     Running   0           4h59m  172.38.232.285  10.104.182.217   <none>           <none>

D:\final_deliver> kubecttl describe customer-care-registry
error: the server doesn't have a resource type "customer-care-registry"

D:\final_deliver> kubecttl describe customer-care-registry-service
error: the server doesn't have a resource type "customer-care-registry-service"

D:\final_deliver>kubecttl get svc -o wide
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE    SELECTOR
customer-care-registry-service      NodePort            172.21.61.147   <none>           5000:32443/TCP   93m    app=customer-care-registry
flask-app                          ClusterIP           172.21.113.253   <none>           5000/TCP         4h47m  app=flask-app
flaskservice1                      ClusterIP           172.21.143.245   <none>           5000/TCP         5h11m  app=flask-app
flaskservice1a                     NodePort            172.21.242.76    <none>           5000:31827/TCP   4h18m  app=flask-app
kubernetes                         ClusterIP           172.21.0.1        <none>           443/TCP          5h47m  <none>

D:\final_deliver>kubecttl get svc -o wide
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE    SELECTOR
customer-care-registry-service      NodePort            172.21.61.147   <none>           5000:32443/TCP   182m    app=customer-care-registry
flask-app                          ClusterIP           172.21.113.253   <none>           5000/TCP         4h57m  app=flask-app
flaskservice1                      ClusterIP           172.21.143.245   <none>           5000/TCP         5h28m  app=flask-app
flaskservice1a                     NodePort            172.21.242.76    <none>           5000:31827/TCP   4h29m  app=flask-app
kubernetes                         ClusterIP           172.21.0.1        <none>           443/TCP          5h56m  <none>

D:\final_deliver>kubecttl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE    IP              NODE              NOMINATED NODE   READINESS GATES
customer-care-registry-57q7q58877-xz9dx  1/1     Running   0           145m    172.38.232.287  10.104.182.217   <none>           <none>
flask-app-798b5b7dcd-lfcd6             1/1     Running   0           7h12m   172.38.232.285  10.104.182.217   <none>           <none>

D:\final_deliver>

```

Org:
Space:

D:\final_deliver>ibmcloud cr login
Logging 'docker' in to 'jp.icr.io'...
Logged in to 'jp.icr.io'.

OK

D:\final_deliver>docker push jp.icr.io/project_ccr/repo:finaldeliverable
The push refers to repository [jp.icr.io/project_ccr/repo]
a4c4c881a223: Layer already exists
836495f54f13: Pushed
e78afbd4b71cb: Layer already exists
2e1cb58aa872: Layer already exists
9fda48ddc568: Layer already exists
428e1f341db7: Layer already exists
8ea8d700cd5d: Layer already exists
136445a1d4d2: Layer already exists
2fbabeba982e: Layer already exists
ee589edde976: Layer already exists
9177197c67d0: Layer already exists
7dbad72b9bd8: Layer already exists
e7897c383c2e: Layer already exists
finaldeliverable: digest: sha256:96e44b178a8628ade0506c340f273603cddb923c32fb0a6af25b0f7a1071ed1a size: 3855

D:\final_deliver>kubectl apply -f kubernetes/flask_deployment.yaml
deployment.apps/customer-care-registry created

D:\final_deliver>kubectl apply -f kubernetes/flask_service.yaml
service/customer-care-registry-service created

D:\final_deliver>kubectl apply -f kubernetes/flask_ingress.yaml
ingress.networking.k8s.io/customer-care-registry-ingress created

D:\final_deliver>kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
customer-care-registry-5b9d87c86-gtrvg	0/1	ImagePullBackOff	0	53s
flask-app-5989b67fb7-6a4hd	1/1	Running	3 (3m27s ago)	6h22m